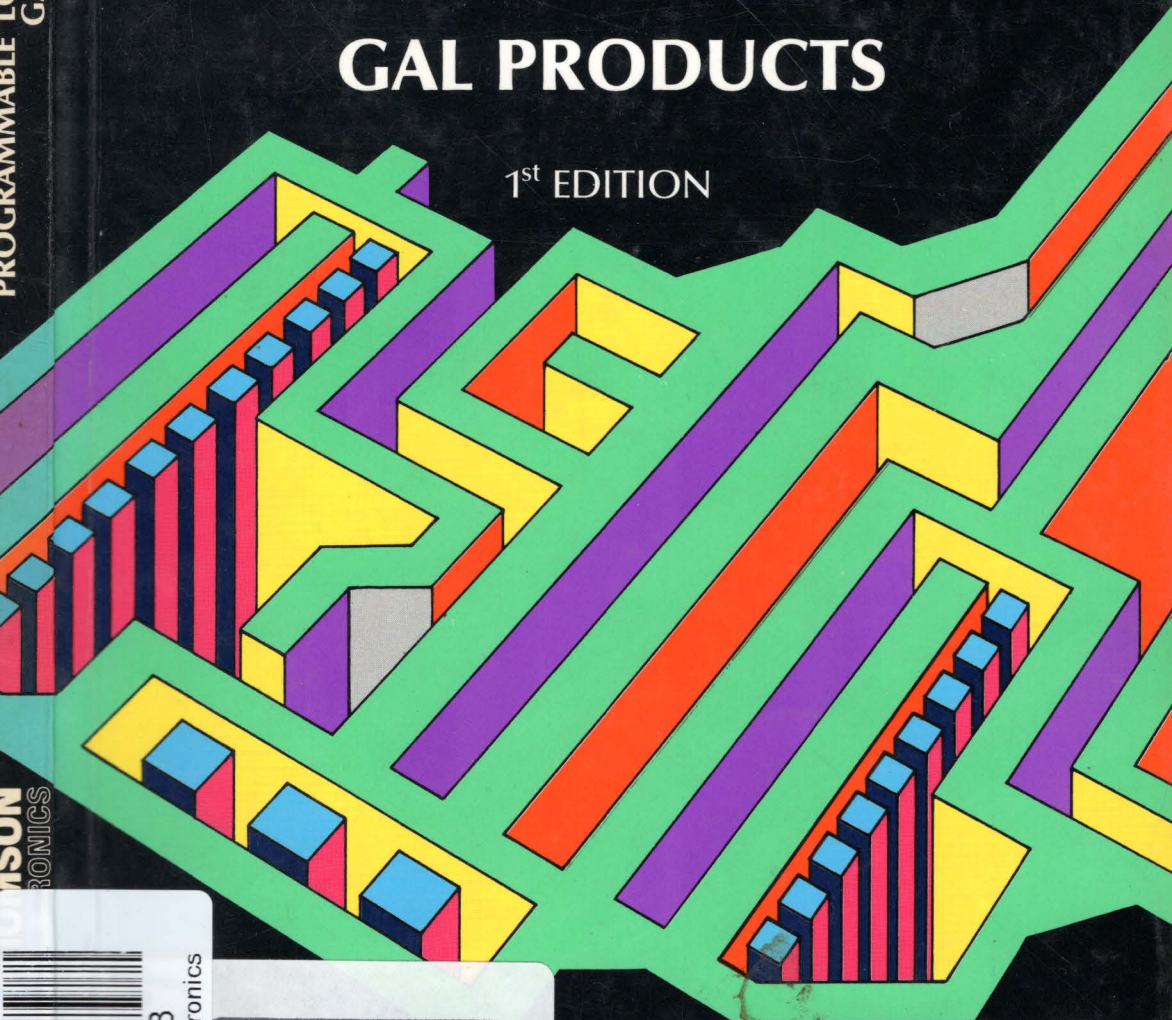


PROGRAMMABLE LOGIC MANUAL
GAL PRODUCTS

PROGRAMMABLE LOGIC MANUAL

GAL PRODUCTS

1st EDITION



THOMSON
RONICS



000533

RYSTON Electronics

RYSTON
ELECTRONICS
spol. s r.o.
Na hřebenech II 1062
147 00 Praha 4

THOMSON
ROELECTRONICS

PROGRAMMABLE LOGIC MANUAL

GAL PRODUCTS

1st EDITION

OCTOBER 1988

USE IN LIFE SUPPORT MUST BE EXPRESSLY AUTHORIZED

SGS-THOMSON' PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF SGS-THOMSON Microelectronics. As used herein:

1. Life support devices or systems are devices or systems which, are intended for surgical implant into the body to support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

TABLE OF CONTENTS

GENERAL INDEX	Page 5
INTRODUCTION	7
DATASHEETS	13
LOGIC CONCEPT	73
DEVELOPMENT TOOLS	91
GAL DEVICE APPLICATIONS	99
TECHNOLOGY AND TECHNICAL OVERVIEW	179
QUALITY AND RELIABILITY	197

GENERAL INDEX

	Page
INTRODUCTION	7
GAL IN PLD SCENARIO	9
DEFINITION OF SYMBOLS AND TERMS USED IN THIS DATA BOOK	11
DATASHEETS	13
GAL16V8	15
GAL20V8	35
GAL16Z8	55
GAL39V18	61
LOGIC CONCEPT	73
LOGIC DESIGN BASIC CONCEPTS	75
GAL CONFIGURATION EXAMPLE	83
DEVELOPMENT TOOLS	91
SOFTWARE AND HARDWARE TOOLS	93
THE DESIGN PROCESS	95
GAL DEVICE APPLICATIONS	99
INTRODUCTION TO GAL APPLICATIONS	100
BASIC GATES	101
BASIC FLIP-FLOP	118
QUAD 1:1 MULTIPLEXER	136
SEVEN SEGMENT DISPLAY DRIVER	147
7-BIT COUNTER	155
COPY PAL PATTERNS INTO GAL DEVICE	168
USING ELECTRONIC SIGNATURE	175
OVERVIEW OF REGISTER PRELOAD	177
TECHNOLOGY AND TECHNICAL OVERVIEW	179
THE GAL PROCESS	189
POWER CONSIDERATION	191
LATCH-UP PROTECTION	194

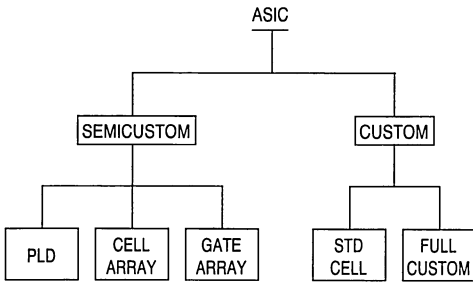
GENERAL INDEX

	Page
QUALITY AND RELIABILITY	197
QUALITY PROGRAM	199
WAFER FAB STANDARD PRODUCTION PROCESS FLOWCHART	202
QUALITY INSPECTIONS/MONITORS DURING WAFER FABRICATION	203
ASSEMBLY STANDARD PRODUCTION PROCESS FLOW CHART	204
QUALITY TESTS DURING ASSEMBLY PROCESS	205
GROUPS B, C AND D RELIABILITY TESTS	206

INTRODUCTION

GAL IN PLD SCENARIO

With complete logic source philosophy, SGS-THOMSON is today committed to serving the market with high volume families and introducing Programmable Logic Devices (PLDs), which show one of the fastest growth rates in the semiconductor market. PLDs in the ASIC scenario: PLDs represent today one of the largest families in the ASIC market.



The devices in the ASIC category offer advantages over other alternatives, in that they perform a function defined by the user that is optimized for a specific application. The ASIC category further divides into other categories of functionally specific devices as shown in the above figure.

The standard cell approach uses pre-configured, pre-tested and pre-characterized logic blocks to construct a custom silicon chip for the designer. The chip is usually designed by hand, using a graphics terminal. The outcome is a fairly efficient logic design that may take weeks or months to complete, while incurring a hefty up-front engineering charge, or non-recurring expense (NRE).

In addition, the custom piece of silicon will take additional weeks, if not months, to manufacture. The development time and complexity of a standard cell design severely impacts the ability to incorporate changes or corrections to the design. In addition, the customer is typically obligated for some minimum production lot size to cover the manufacturer's expenses. Since most logic design is subject to revision during the debug phase, the time and dollar penalties of the standard cell approach make this a relatively high risk.

The gate array approach has gained extensive market recognition as a more optimal "bridge" between the standard cell (full custom) and programmable alternatives. The gate array is a pre-manufactured silicon matrix that awaits only a custom interconnected pattern to establish functionality. The designer can choose from NAND gates, flip-flops, and various types of buffers to construct the logic.

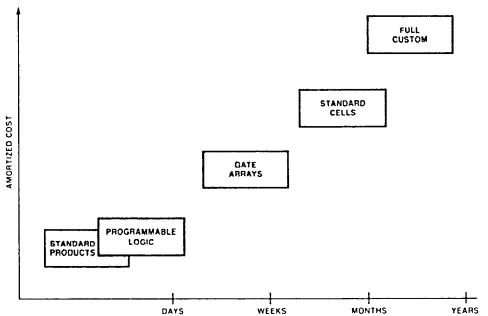
The flexibility of a gate array is less than that of a standard cell device, since the user must interconnect existing structures. However, since the device usually has many pins -68 or more it offers greater logic functionality than a typical PLD. This increased functionality, however, comes at the penalty of lower speed performance.

Since the gate array relies on only one or two custom mask layers, the wafer fabrication can be done much more quickly than with a standard cell. The turn time from design completion to final chip for a gate array is, at best, 4 to 8 weeks. Although not as costly up front as a standard cell, there is still an up front development cost, a minimum lot size, and a long and costly cycle for logic changes. These features make this, too, a risky approach.

Advantages of PLDs

The previous alternatives are ideal choices for high volume applications where, once debugged, the design is not subject to change. The average customer, however, uses hundreds to thousands of devices of a given logic pattern, each year. He cannot afford the NRE of a gate array or standard cell, since the volume is not high enough to amortize this expense. Figure 1 shows the cost and development time relationships of the various design alternatives.

Figure 1. Costs vs. Development Time



The PLD offers a solution to these woes. The low unit cost, simple, but powerful and affordable development tools, flexibility, high performance, and proven reliability of these devices results in a more cost-efficient, higher-performance, lower-risk, and more timely design cycle.

Programmable logic is ideal for simplifying the design process, because the designer can implement the exact logic function whenever and wherever required. Programmable logic offers more efficient utilization, as well as reduced chip count, by simplifying the lay-out process at both conceptual and implementation stage.

Consequently PLDs contributes to increase substantially system reliability. It has been statistically demonstrated that systems with higher levels of integration such those as designed with programmable logic, have much higher reliability than equivalent systems designed with many low density standard components.

GAL

SGS-THOMSON has decided to enter the PLD market by producing the GAL (Generic Array Logic) family of devices. GAL's are ideal devices among PLD's for several reasons:

- GAL devices are fabricated using very high speed Electrically Erasable CMOS technology which offers the highest degree of testability and quality of any process technology. In fact AC, DC and functionality can be 100% tested and that guarantees 100% programming and functional yield to the customer with no further board rework. With bipolar technologies, complete testing is not possible and manufacturers must rely on complex schemes using test rows and columns to simulate and correlate devices performances, since the fuse array cannot be tested prior to programming.

Consequently due to incomplete test, failures at the customer location are possible with no recovery possibility since misprogrammed parts must be discarded. Besides, instant erasability

makes GALs ideal for prototyping and for unforecasted design changes.

- GAL devices speeds are as fast as any other bipolar programmable logic devices except ECL, but they have the low power consumption of CMOS.
- GAL devices utilize the OUTPUT LOGIC MACROCELL (OLMC) which allows the user to configure outputs as needed and to replace several other programmable logic devices and low density gate arrays.

The main advantage of GAL devices comes from their intrinsic "genericity" that allows the user to define the architecture and functionality of each output and also has advantages at the shop floor level: users can put in inventory one generic GAL type instead of many different PAL device types; this will not only save money, but also minimize the paper work, reduce manufacturing flow because the handling process is simplified, reduce the risk of running out of inventory. For example, the GAL 16V8 can replace 21 different bipolar PALs.

GAL Development Tools

GAL devices have been developed to support the philosophy that users should not be required to purchase special development tools. GALs are in fact supported by existing programmable logic development tools and device programmers.

SGS-THOMSON doesn't offer any special software development package, this shouldn't represent a limitation because user is not required to purchase and maintain a dedicated software for each programmable devices being used.

Software packages such as ABEL from DATA I/O and CUPL from PC ad offer generic development support for all programmable logic devices and SGS-THOMSON can supply all the necessary support to work with these software tools. GALs are in the same way supported by many "devices programmers" that ensure the highest quality.

DEFINITION OF SYMBOLS AND TERMS USED IN THIS DATA BOOK

DC Symbols and definitions

VOLTAGES - All voltages are referred to ground. Negative voltage limits are specified as absolute values (i. e. $-10V$ is greater than $-1.0V$).

Symbol	Parameter	Description
V_{CC}	Supply Voltage	The range of power supply voltage over which the device is guaranteed to operate within specified limits.
V_{IH}	Input HIGH Voltage	The range of input voltages that represents a logic HIGH in the system.
$V_{IH}(\text{min})$	Minimum Input HIGH Voltage	The minimum allowed input HIGH in a logic system. This value represents the guaranteed input HIGH threshold.
V_{IL}	Input LOW Voltage	The range of input voltages that represents a logic LOW in the system.
$V_{IL}(\text{max})$	Maximum Input LOW Voltage	The maximum allowed input LOW in a system. This value represents the guaranteed input LOW threshold for the device.
$V_{OH}(\text{min})$	Output HIGH Voltage	The minimum voltage at an output terminal for the specified output current I_{OH} and the minimum value of V_{CC} .
$V_{OL}(\text{max})$	Output LOW Voltage	The maximum voltage at an output terminal sinking the maximum specified LOW current I_{OL} .

CURRENTS - Positive current is defined as conventional current flow into a device. Negative current is defined as conventional current flow out of a device. All current limits are specified as absolute values.

Symbol	Parameter	Description
I_{CC}, I_{SB}	Supply Current	The current flowing into the V_{CC} supply terminal of a circuit with the specified inputs conditions and the outputs open. When not specified, input conditions are chosen to guarantee worst case operation.
I_F	Input Leakage Current	The current flowing into an input when the maximum allowed voltage is applied to the input. This parameter guarantees the minimum breakdown voltage for the input.
I_{IH}	Input HIGH Current	The current flowing into an input when a specified HIGH level voltage is applied to that input.
I_{IL}	Input LOW Current	The current flowing out of an input when a specified LOW voltage is applied to the input.
I_{OH}	Output HIGH Current	The current flowing out of an output which is in the HIGH state.
I_{OL}	Output LOW Current	The current flowing into an output which is in the LOW state.
I_{OS}	Output Short Circuit Current	The current flowing out of an output which is in the HIGH state when that output is short circuit to ground.
I_{BZH}, I_{FZH}	Bidirectional pin Leakage Current HIGH	The current flowing into a disabled 3-state output with a specified HIGH output voltage applied.
I_{BZL}, I_{FZL}	Bidirectional pin Leakage Current LOW	The current flowing out of a disabled 3-state output with a specified LOW output voltage applied.

AC SWITCHING PARAMETERS

Symbol	Parameter	Description
f_{MAX}	Maximum Clock Frequency	The maximum input frequency at a clock input for predictable performance. Above this frequency the device may cease to function.
T_{DVQV1} , T_{CHQV}	Propagation Delay Time	The time between the specified reference points on the input and output waveforms with the output changing from the defined LOW level to the defined HIGH level.
T_{DVQV1} , T_{CHQV}	Propagation Delay Time	The time between the specified reference points on the input and output waveforms with the output changing from the defined HIGH level to the defined LOW level.
T_{DVQZ} , T_{GHQZ}	Output Disable Time from HIGH Level of a 3-State Output	The delay time between the specified reference points on the input and output voltage waveforms with the 3-state output changing from the HIGH level to a high impedance "off" state.
T_{DVQZ} , T_{GHQZ}	Output Disable Time from LOW Level of a 3-State Output	The delay time between the specified reference points on the input and output voltage waveforms with the 3-state output changing from the LOW level to a high impedance "off" state.
T_{DVQV2} , T_{GLQV}	Output Enable Time to a HIGH Level of a 3-State Output	The delay time between the specified reference points on the input and output voltage waveforms with the 3-state output changing from a high impedance "off" state to the HIGH level.
T_{DVQV2} , T_{GLQV}	Output Enable Time to a LOW Level of a 3-State Output	The delay time between the specified reference points on the input and output voltage waveforms with the 3-state output changing from a high impedance "off" state to the LOW level.
T_{CHDX}	Hold Time	The interval immediately following the active transition of the timing pulse (usually the clock pulse) or following the transition of the control input to its latching level, during which interval the data to be recognized must be maintained at the input to ensure its continued recognition. A negative setup time indicates that the correct logic level may be released prior to the active transition of the timing pulse and still be recognized.
T_{DVCH}	Setup Time	The interval immediately preceding the active transition of the timing pulse (usually the clock pulse) or preceding the transition of the control input to its latching level, during which interval the data to be recognized must be maintained at the input to ensure its recognition. A negative setup time indicates that the correct logic level may be initiated sometimes after the active transition of the timing pulse and still be recognized.
T_{CHCL} , T_{CLCH} , T_{CHCH}	Pulse Width	The time between the specified reference points on the leading and trailing edges of a pulse.

Note: 1 – Refer also to switching test conditions

2 – 3 State levels are measured 0.5V from steady-state active level.

DATASHEETS

E²CMOS PROGRAMMABLE LOGIC DEVICE

PRELIMINARY DATA

- **ELECTRICALLY ERASABLE CELL TECHNOLOGY**
 - Reconfigurable Logic
 - Reprogrammable Cells
 - Guaranteed 100% Yields
- **HIGH PERFORMANCE E²CMOS TECHNOLOGY**
 - Low Power : 90/70mA Max Active/Standby (Half Power)
45/35mA Max Active/Standby (Quarter Power)
 - High Speed: 15 to 25ns Access Max (Half Power)
20 to 35ns Access Max (Quarter Power)
- **EIGHT OUTPUT LOGIC MACROCELLS**
 - Maximum Flexibility for Complex Logic Designs
 - Also Emulates 20-pin PAL* Devices with Full Function/Fuse Map/Parametric Compatibility
- **PRELOAD AND POWER-ON RESET OF ALL REGISTERS**
 - 100% Functional Testability
- **HIGH SPEED PROGRAMMING ALGORITHM**
- **SECURITY CELL PREVENTS COPYING LOGIC**
- **DATA RETENTION EXCEEDS 20 YEARS**

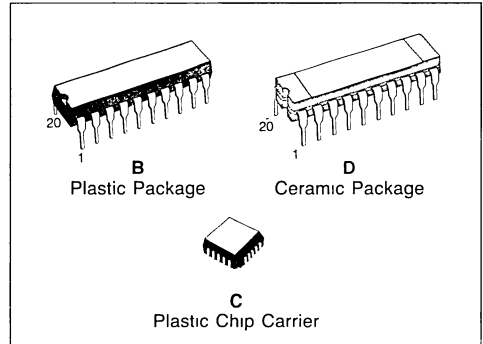
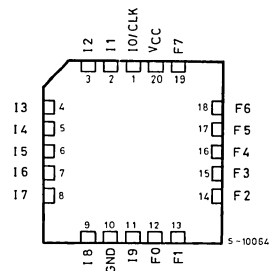
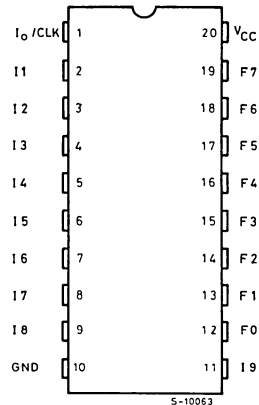
DESCRIPTION

The GAL* 16V8 E²CMOS device combines a high performance CMOS process with electrically erasable floating gate technology. This programmable memory technology applied to array logic provides designers with reconfigurable logic and bipolar performance at significantly reduced power levels. The 20-pin GAL16V8 features 8 programmable Output Logic Macrocells (OLMCs) allowing each output to be configured by the user. Additionally, the GAL16V8 is capable of emulating, in a functional/fuse map/parametric compatible device, all 20 pin PAL device architectures (excluding XOR devices).

Programming is accomplished using readily available hardware and software tools.

SGS-THOMSON guarantees 100 erase/write cycles and data retention exceeds 20 years.

Unique test circuitry and reprogrammable cells allows complete AC,DC cell and functionality testing during manufacture. Therefore, SGS-THOMSON guarantees 100% field programmability and functionality of the GAL devices. In addition, electronic signature is available to provide positive device ID. A security circuit is built-in, providing proprietary designs with copy protection.


PIN CONNECTIONS (top view)


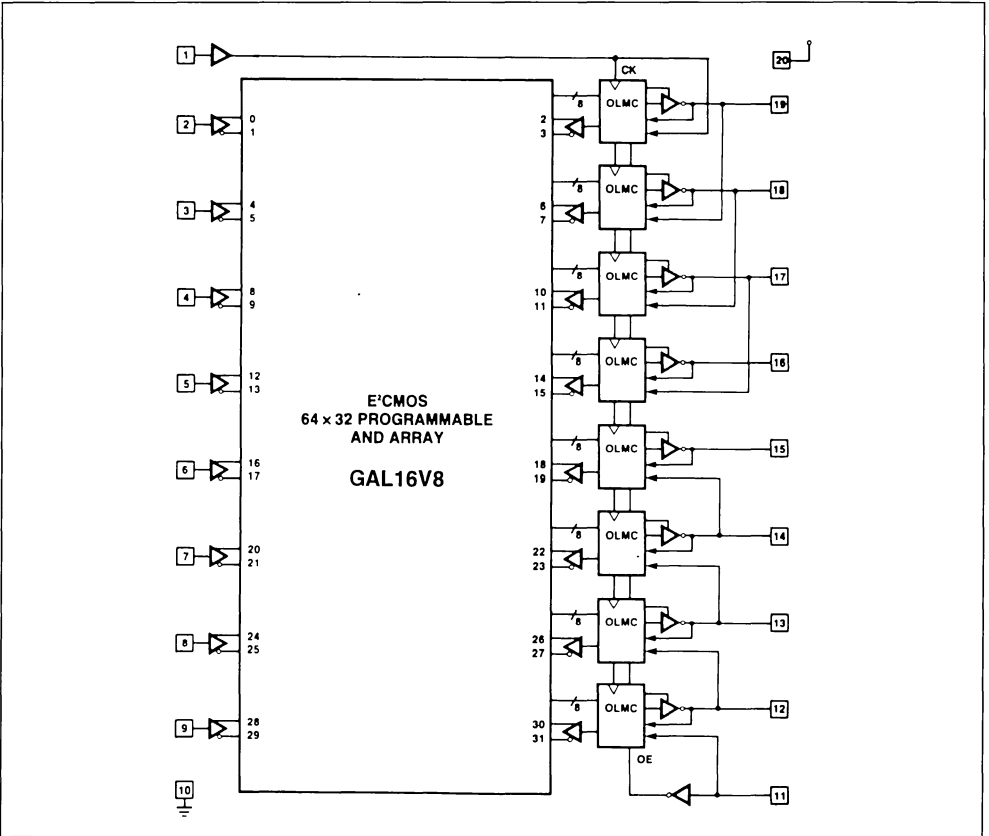
PIN NAMES

I0-I15	INPUT
CK	CLOCK INPUT
B0-B15	BIDIRECTIONAL
F0-F7	OUTPUT
OE	OUTPUT ENABLE
V _{CC}	POWER (+5V)
GND	GROUND

GAL16V8 EMULATING PAL DEVICES

10/CLK	V _{CC}	V _{CC}	V _{CC}	V _{CC}	V _{CC}	V _{CC}	V _{CC}	V _{CC}
11	F7	I11	I13	I15	F7	B1	B3	F1
12	F6	F5	I12	I14	F6	F5	B2	B5
13	F5	F4	F3	I13	F5	F4	F3	B4
14	F4	F3	F2	F1	F4	F3	F2	B3
15	F3	F2	F1	F0	F3	F2	F1	B2
16	F2	F1	F0	I12	F2	F1	F0	B1
17	F1	F0	I11	I11	F1	F0	B1	B0
18	F0	I10	I10	I10	F0	B0	B0	F0
GND	I9	I9	I9	I9	OE	OE	OE	I9
	10L8	12L6	14L4	16L2	16R8	16R6	16R4	16L8
	10H8	12H6	14H4	16H2	16RP8	16RP6	16RP4	16H8
	10P8	12P6	14P4	16P2				16P8
GAL16V8								

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	- 0.5 to +7	V
V_I	Input Voltage Applied	- 2.5 to $V_{CC} + 1.0$	V
V_O	Off-State Output Voltage Applied	- 2.5 to $V_{CC} + 1.0$	V
T_{stg}	Storage Temperature	- 65 to 125	°C

Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specification).

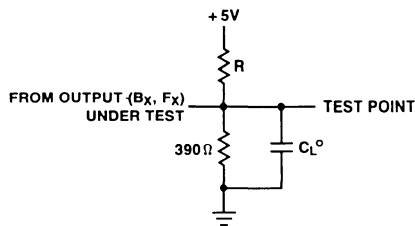
OPERATING RANGE

Symbol	Parameter	Temperature Range						Unit
		Military			Commercial			
		Min.	Typ.	Max.	Min.	Typ.	Max.	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
T_A	Ambient Temperature				0		70	°C
T_C	Case Temperature	- 55		125				°C

SWITCHING TEST CONDITIONS

Input Pulse Level	GND to 3.0V
Input Rise and Fall Times	3ns 10%-90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See figure

3-state levels are measured 0.5V from steady-state active level.



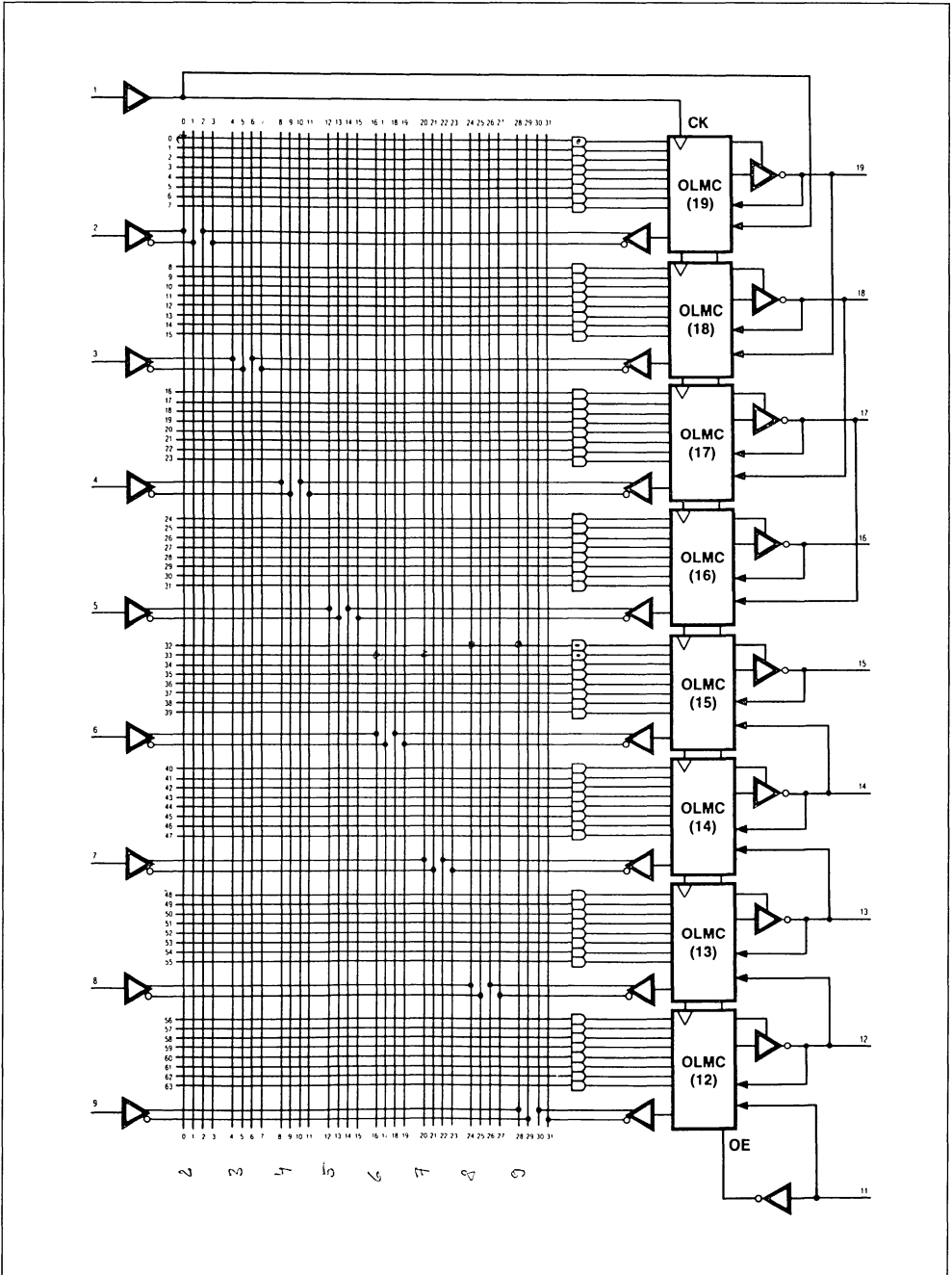
(°) C_L INCLUDES JIG AND PROBE TOTAL CAPACITANCE

CAPACITANCE ($T_A = 25^\circ\text{C}$, $f = 1.0\text{MHz}$, $V_{CC} = 5\text{V}$)

Symbol	Parameter	Test Conditions	Maximum*	Units
C_I	Input Capacitance	$V_I = 2\text{V}$	12	pF
C_F	Output Capacitance	$V_F = 2\text{V}$	15	pF
C_B	Bidirectional Pin Cap	$V_B = 2\text{V}$	15	pF

* Guaranteed but not 100% tested

LOGIC DIAGRAM



ELECTRICAL CHARACTERISTICS OVER OPERATING CONDITIONS

- ADVANCE DATA - Quarter Power GAL16V8

Symbol	Parameter	Test Conditions	Temp. Range	Min.	Max.	Unit
I_{IH}, I_{IL}	Input Leakage Current	$GND \leq V_{IN} \leq V_{CC} \text{ MAX}$		—	± 10	μA
I_{BZH}, I_{BZL}	Bidirectional Input Leakage Current	$GND \leq V_{IN} \leq V_{CC} \text{ MAX}$		—	± 10	μA
I_{FZL}, I_{FZH}	Output Pin Leakage Current	$GND \leq V_{IN} \leq V_{CC} \text{ MAX}$		—	± 10	μA
I_{CC}	Operating Power Supply Current	$f = 15\text{MHz}$ $V_{CC} = V_{CC} \text{ MAX}$	COM'L	—	45	mA
			MIL	—	50	mA
$I_{OS}(\text{°})$	Output Short Circuit Current	$V_{CC} = 5.0\text{V}$ $V_{OUT} = GND$		-30	-130	mA
I_{SB}	Standby Power Supply Current	$V_{CC} = V_{CC} \text{ MAX}$	COM'L	—	35	mA
			MIL	—	45	mA
V_{OL}	Output Low Voltage $V_{CC} = V_{CC} \text{ MIN}$	$I_{OL} = 24\text{mA}$	COM'L	—	0.5	V
		$I_{OL} = 12\text{mA}$	MIL	—	0.5	V
V_{OH}	Output High Voltage $V_{CC} = V_{CC} \text{ MIN}$	$I_{OH} = -3.2\text{mA}$	COM'L	2.4	—	V
		$I_{OH} = -2.0\text{mA}$	MIL	2.4	—	V
V_{IH}	Input High Voltage			2.0	$V_{CC} + 1$	V
V_{IL}	Input Low Voltage			—	0.8	V

HALF POWER GAL16V8

Symbol	Parameter	Test Conditions	Temp. Range	Min.	Max.	Unit
I_{IH}, I_{IL}	Input Leakage Current	$GND \leq V_{IN} \leq V_{CC} \text{ MAX}$		—	± 10	μA
I_{BZH}, I_{BZL}	Bidirectional Pin Leakage Current	$GND \leq V_{IN} \leq V_{CC} \text{ MAX}$		—	± 10	μA
I_{FZL}, I_{FZH}	Output Pin Leakage Current	$GND \leq V_{IN} \leq V_{CC} \text{ MAX}$		—	± 10	μA
I_{CC}	Operating Power Supply Current	$f = 15\text{MHz}$ $V_{CC} = V_{CC} \text{ MAX}$	COM'L	—	90	mA
			MIL	—	90	mA
$I_{OS}(\text{°})$	Output Short Circuit Current	$V_{CC} = 5.0\text{V}$ $V_{OUT} = GND$		-30	-130	mA
I_{SB}	Standby Power Supply Current	$V_{CC} = V_{CC} \text{ MAX}$	COM'L	—	70	mA
			MIL	—	70	mA

(°) One output at a time for a maximum duration of one second.

ELECTRICAL CHARACTERISTICS OVER OPERATING CONDITIONS (Continued)

Symbol	Parameter	Test Conditions	Temp. Range	Min.	Max.	Unit
V _{OL}	Output Low Voltage V _{CC} = V _{CC} MIN	I _{OL} = 24mA	COM'L	—	0.5	V
		I _{OL} = 12mA	MIL	—	0.5	V
V _{OH}	Output High Voltage V _{CC} = V _{CC} MIN	I _{OH} = -3.2mA	COM'L	2.4	—	V
		I _{OH} = -2.0mA	MIL	2.4	—	V
V _{IH}	Input High Voltage			2.0	V _{CC} + 1	V
V _{IL}	Input Low Voltage			—	0.8	V

SWITCHING CHARACTERISTICS OVER OPERATING CONDITIONS

ADVANCE DATA - Quarter Power GAL16V8

Symbol	Parameter	TEMPERATURE RANGE								Units	Test Conditions (°)		
		0°C to 70°C						-55 to 125°C			R (Ω)	C _L (pF)	
		16V8-20		16V8-25		16V8-35		16V8-30					
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.				
T _{DVQV1}	Delay from Input to active output (°°°)	—	20	—	25	—	35	—	30	ns	200	50	
T _{DVQV2}	Product Term Enable Access Time to Active Output	—	20	—	25	—	35	—	30	ns	Active High R = ∞ Active Low R = 200	50	
T _{DVQZ} (°°)	Product Term Enable to Outputs Off	—	20	—	25	—	35	—	30	ns	From V _{OH} R = ∞ From V _{OL} R = 200	5	
T _{GHQZ} (°°)	\overline{OE} Output Enable High to Output Off	—	18	—	20	—	25	—	25	ns	From V _{OH} R = ∞ From V _{OL} R = 200	5	
T _{GLQV}	\overline{OE} Output Enable Access Time	—	18	—	20	—	25	—	25	ns	Active High R = ∞ Active Low R = 200	50	
T _{CHQV}	Clock High to Output Valid Access Time	—	15	—	15	—	25	—	20	ns	200	50	
T _{DVCH}	Input or Feedback Data Minimum Setup Time	—	15	—	20	—	30	—	25	ns	—	—	
T _{CHDX}	Input or Feedback Data Minimum Hold Time	—	0	—	0	—	0	—	0	ns	—	—	
T _{CHCH}	Minimum Clock Period (T _{DVCH} + T _{CHQV})	—	30	—	35	—	55	—	45	ns	—	—	
T _{CHCL}	Minimum Clock Width HIGH	—	12	—	15	—	20	—	15	ns	—	—	
T _{CLCH}	Minimum Clock Width LOW	—	12	—	15	—	20	—	15	ns	—	—	
f _{MAX}	Maximum Frequency	SYNCH.	33.3	—	28.5	—	18.1	—	22.2	—	MHz	200	50
		ASYNCH.	50.0	—	40.0	—	28.5	—	33.3	—			

(°) Refer also to switching test conditions. (°°) 3-state levels are measured 0.5V from steady-state active level.

(°°°) XOR (n) = 1 (active high)

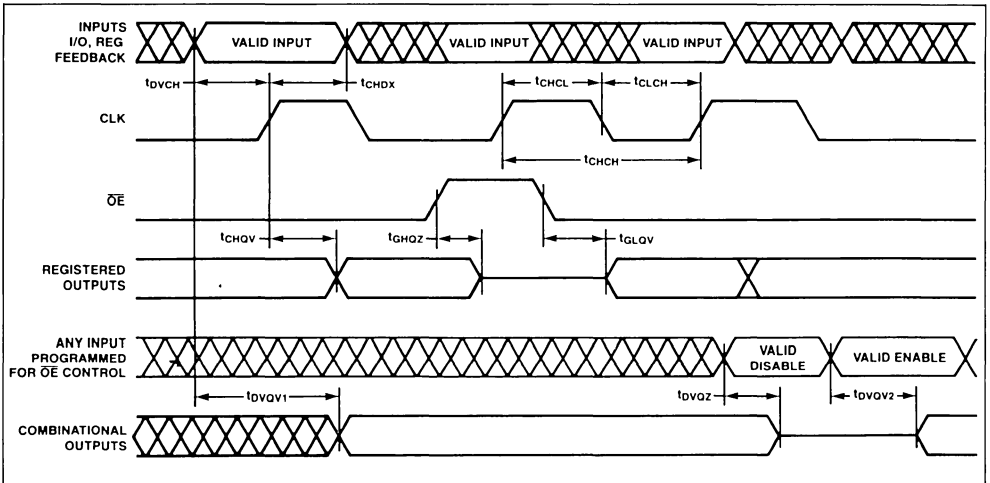
SWITCHING CHARACTERISTICS OVER OPERATING CONDITIONS
HALF POWER GAL16V8

Symbol	Parameter	TEMPERATURE RANGE										Units	Test Conditions(*)		
		0°C to 70°C					-55 to 125°C						R (Ω)	C _L (pF)	
		16V8-15		16V8-20		16V8-25		16V8-20		16V8-30					
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.				
T _{DVQV1}	Delay from Input to active output (°°°)	—	15	—	20	—	25	—	20	—	30	ns	200	50	
T _{DVQV2}	Product Term Enable Access Time to Active Output	—	15	—	20	—	25	—	20	—	30	ns	Active High R = ∞ Active Low R = 200	50	
T _{DVQZ} (°°)	Product Term Enable to Outputs Off	—	15	—	20	—	25	—	20	—	30	ns	From V _{OH} R = ∞ From V _{OL} R = 200	5	
T _{GHQZ} (°°)	\overline{OE} Output Enable High to Output Off	—	15	—	18	—	20	—	18	—	25	ns	From V _{OH} R = ∞ From V _{OL} R = 200	5	
T _{GLQV}	\overline{OE} Output Enable Access Time	—	15	—	18	—	20	—	18	—	25	ns	Active High R = ∞ Active Low R = 200	50	
T _{CHQV}	Clock High to Output Valid Access Time	—	12	—	15	—	15	—	15	—	20	ns	200	50	
T _{DVCH}	Input or Feedback Data Minimum Setup Time	—	12	—	15	—	20	—	15	—	25	ns	—	—	
T _{CHDX}	Input or Feedback Data Minimum Hold Time	—	0	—	0	—	0	—	0	—	0	ns	—	—	
T _{CHCH}	Minimum Clock Period (T _{DVCH} + T _{CHQV})	—	24	—	30	—	35	—	30	—	45	ns			
T _{CHCL}	Minimum Clock Width HIGH	—	10	—	12	—	15	—	12	—	15	ns	—	—	
T _{CLCH}	Minimum Clock Width LOW	—	10	—	12	—	15	—	12	—	15	ns			
f _{MAX}	Maximum Frequency	SYNCH.	41.6	—	33.3	—	28.5	—	33.3	—	22.2	—	MHz	200	50
		ASYNCH.	66.6	—	50.0	—	40.0	—	50.0	—	33.3	—			

(°) Refer also to switching test conditions. (°°) 3-state levels are measured 0.5V from steady-state active level.

(°°°) XOR (n) = 1 (active high)

SWITCHING WAVEFORMS



OUTPUT LOGIC MACROCELL (OLMC)

The following discussion pertains to configuring the Output Logic Macrocell. It should be noted that actual implementation is accomplished by development software/hardware and is completely transparent to the user.

The outputs of the AND array are fed into an OLMC, where each output can be individually set to active high or active low, with either combinational (asynchronous) or registered (synchronous) configurations. A common output enable can be connected to all outputs, or separate inputs or product terms can be used to provide individual output enable control. The Output Logic Macrocell provides the designer with maximal output flexibility in matching signal requirements, thus providing more functions than possible with existing 20-pin PAL devices.

The various configurations of the Output Logic Macrocell are controlled by programming certain cells (SYN, AC0, AC1(n) and the XOR(n) polarity bits) within the 82-bit Architecture Control Word. The SYN bit determines whether or not a device will have registered output capability or will have purely combinational outputs. It also replaces the AC0 bit in the two outermost macrocells, OLMC(12) and OLMC(19). When first setting up the device architecture, this is the first bit to choose.

Architecture Control bit AC0 and the eight AC1(n) bits direct the outputs to be wired always on, always off (as an input), have a common OE term

(pin 11), or to be three-state controlled separately from a product term. The Architecture Control bits also determine the source of the array feedback term through the FMUX, and select either combinational or registered outputs.

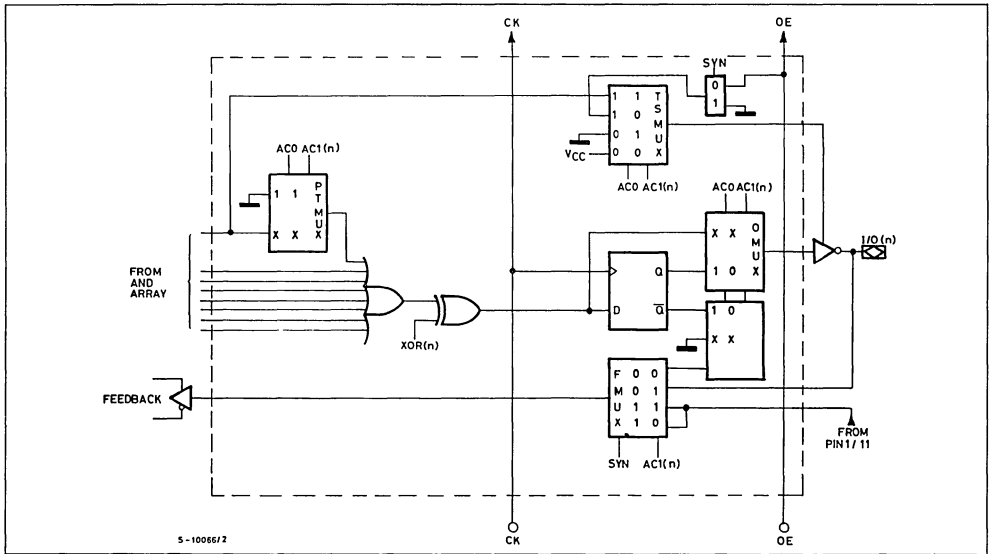
The eight macrocell configurations are shown in each of the macrocell equivalent diagrams, (see fig. 1 to 8 and table 1 and 2). Three of these configurations are not supported by any software development tools. In fact, they are redundant cases included in other configurations. In particular:

- Disabled Output (see fig. 2): the OLMC is completely disabled (nor output neither input function).
- Dedicated input mode (see fig. 7): it is a restrictive case of the other Dedicated input mode configuration shown in fig. 6.
- Dedicated combinational output (see fig. 8): it is a restrictive case of the other Dedicated combinational output shown in fig. 5.

Therefore, if it is strictly necessary to use the previous mentioned configurations, a manual bit-by-bit programming using the software package utilities must be implemented.

In all cases, the eight XOR(n) bits individually determine each output's polarity. The truth table associated with each diagram shows the bit values of the SYN, AC0, and AC1(n) that set the macrocell to the configuration shown.

GAL16V8 Output Logic Macrocell: Pin 12 and 19



GAL16V8 Output Logic Macrocell: Pin 13 to 18

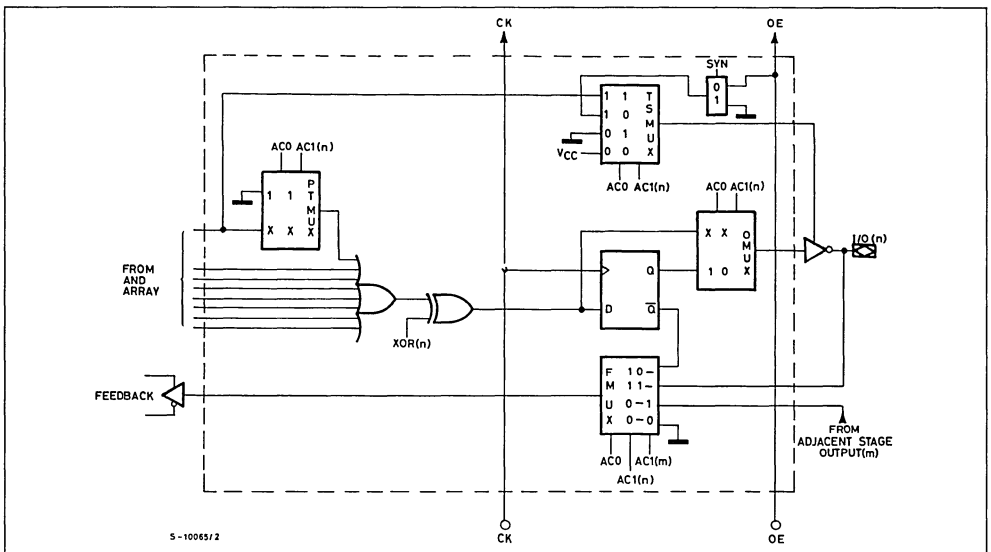


TABLE 1 - ARCHITECTURE OLMC (Pin 12 and 19)

SYN	AC0	AC1(n)	P T MUX	T S MUX	O MUX	F MUX
0	0	0	8 PT	Active OUT	Comb. OUT	No Feedback
0	0	1	Don't Care	3-State OUT	Don't Care	Input Mode
0	1	0	8 PT	Common OE	Regis. OUT	Direct Registered Feedback
0	1	1	7 PT	Single OE (PT)	Comb. OUT	Direct Comb. Feedback or Input
1	0	0	8 PT	Active OUT	Comb. OUT	Pin 1(19) / 11(12)
1	0	1	Don't Care	3-State OUT	Don't Care	Pin 1(19) / 11(12)
1	1	0	Don't Care	3-State OUT	Don't Care	Pin 1(19) / 11(12)
1	1	1	7 PT	Single OE (PT)	Comb. OUT	Pin 1(19) / 11(12)

} Pins 1 and 11
Become Data Inputs

XOR (n)	Output Polarity
0	ACTIVE LOW
1	ACTIVE HIGH

TABLE 2 - ARCHITECTURE OLMC (Pin 13 to 18)

SYN	AC0	AC1(n)	AC1(m)	P T MUX	T S MUX	O MUX	F MUX
0	0	0	0	8 PT	Active OUT	Comb. OUT	No Feedback
0	0	0	1	8 PT	Active OUT	Comb. OUT	Adjacent Macrocell
0	0	1	0	Don't Care	3-State OUT	Don't Care	No Feedback
0	0	1	1	Don't Care	3-State OUT	Don't Care	Adjacent Macrocell
0	1	0	0	8 PT	Common OE	Regis. OUT	Direct Registered Feedback
0	1	0	1	8 PT	Common OE	Regis. OUT	Direct Registered Feedback
0	1	1	0	7 PT	Single OE (PT)	Comb. OUT	Direct Comb. Feedback or Input
0	1	1	1	7 PT	Single OE (PT)	Comb. OUT	Direct Comb. Feedback or Input
1	0	0	0	8 PT	Active OUT	Comb. OUT	No Feedback
1	0	0	1	8 PT	Active OUT	Comb. OUT	Adjacent Macrocell
1	0	1	0	Don't Care	3-State OUT	Don't Care	No Feedback
1	0	1	1	Don't Care	3-State OUT	Don't Care	Adjacent Macrocell
1	1	0	0	Don't Care	3-State OUT	Don't Care	No Feedback
1	1	0	1	Don't Care	3-State OUT	Don't Care	No Feedback
1	1	1	0	7 PT	Single OE (PT)	Comb. OUT	Direct Comb. Feedback or Input
1	1	1	1	7 PT	Single OE (PT)	Comb. OUT	Direct Comb. Feedback or Input

XOR (n)	OUTPUT POLARITY
0	ACTIVE LOW
1	ACTIVE HIGH

Fig. 1 - Combinational Output

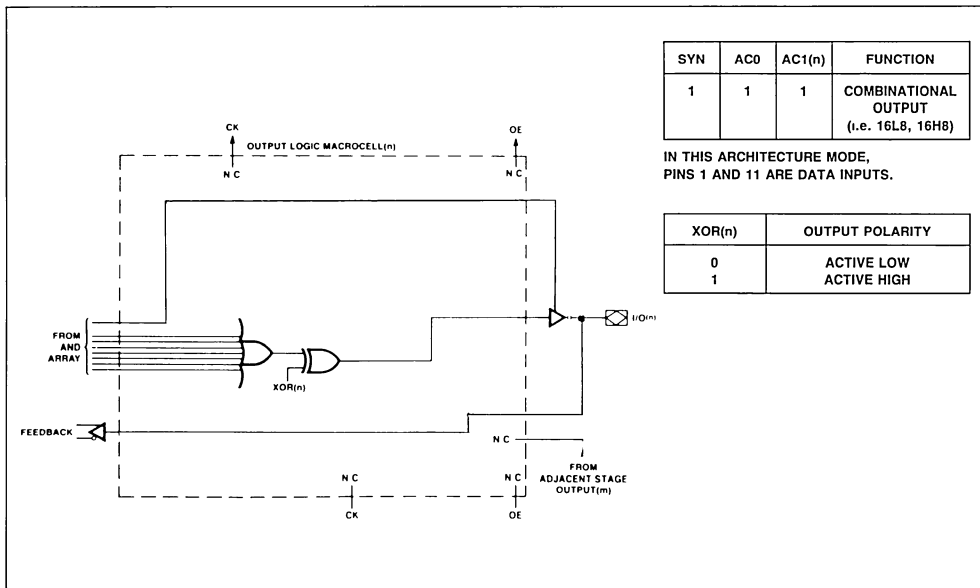


Fig. 2 - Disabled Output

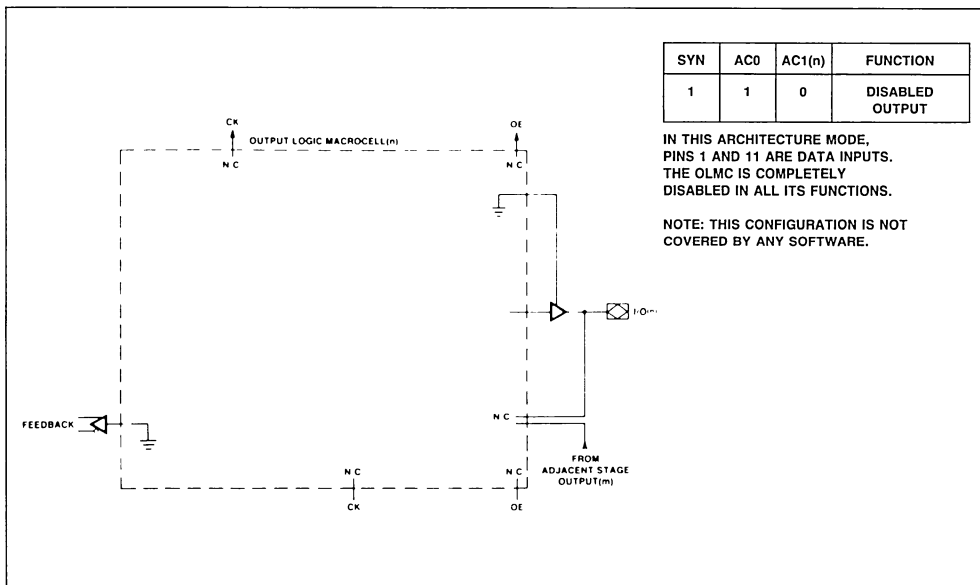


Fig. 3 - Combinational Output in a Registered Device

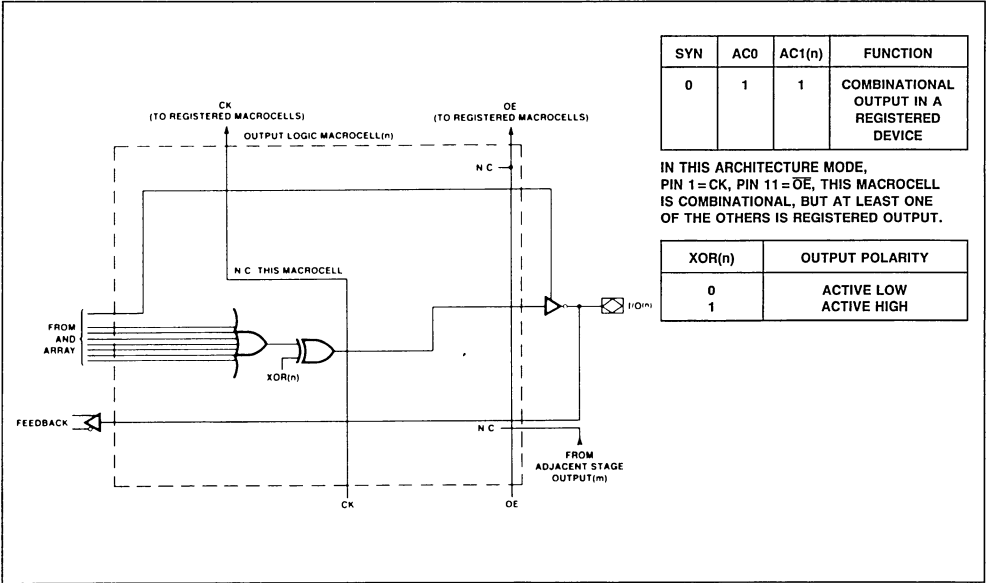


Fig. 4 - Registered Active High or Low Output

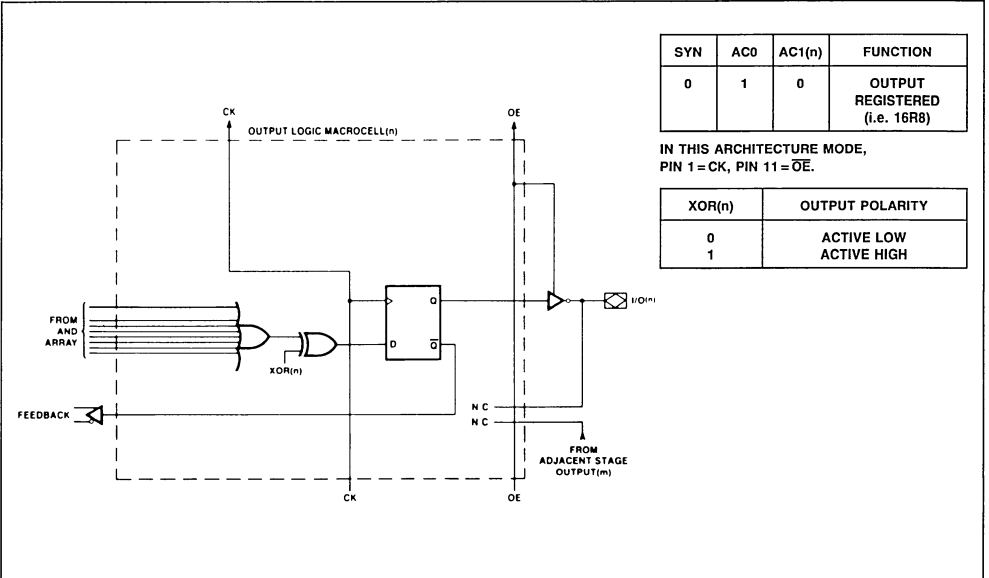


Fig. 5 - Dedicated Combinational Output

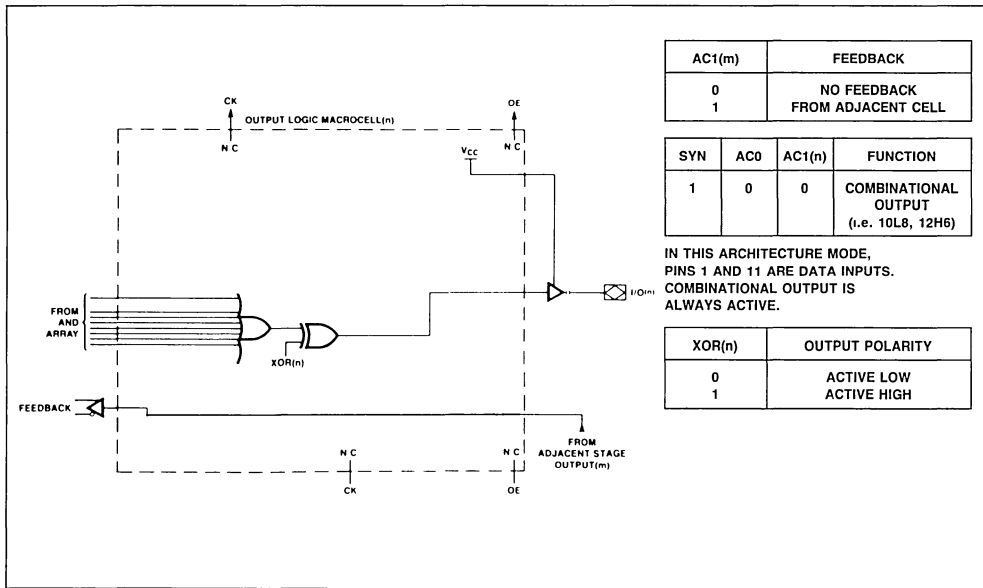


Fig. 6 - Dedicated Input Mode

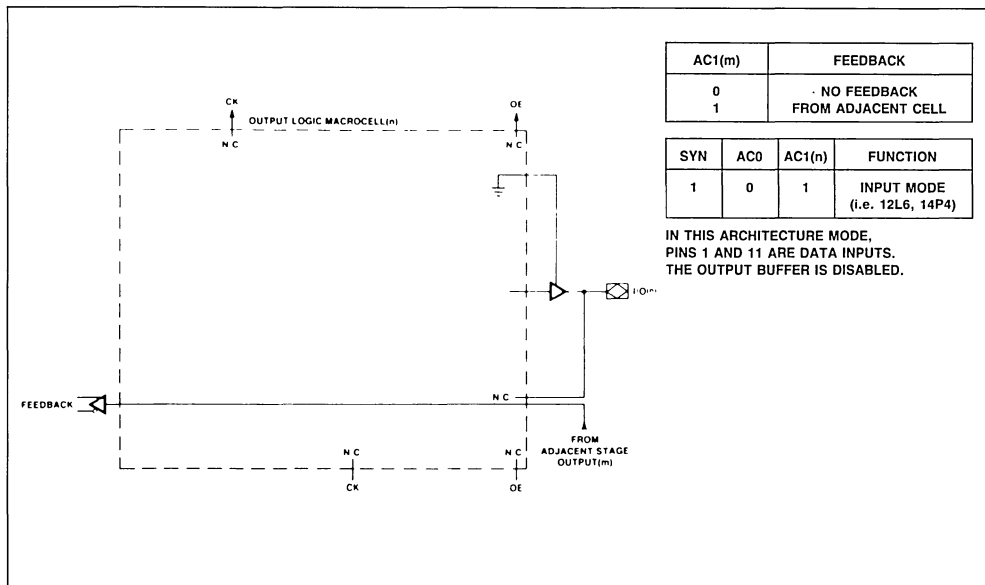


Fig. 7 - Dedicated Input Mode

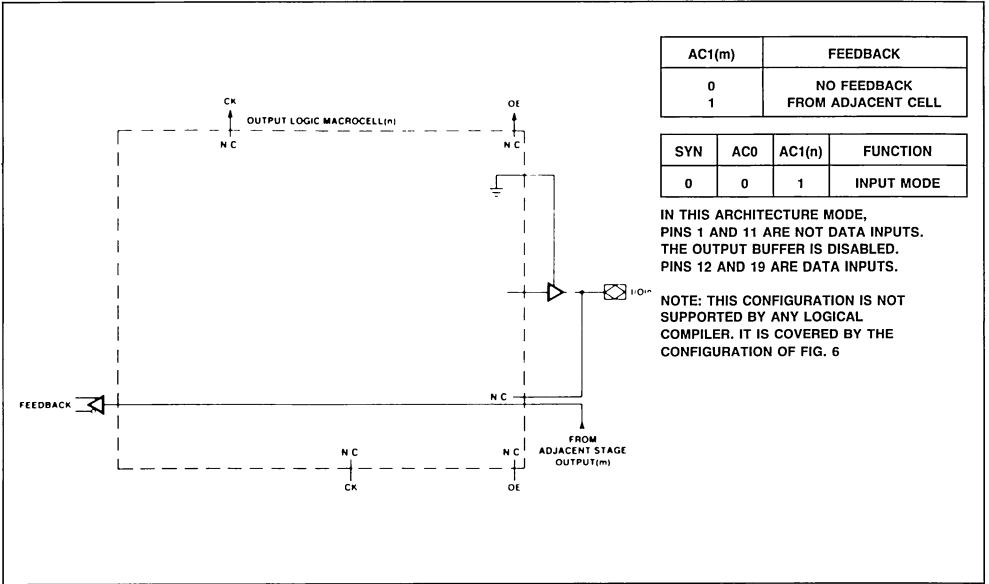
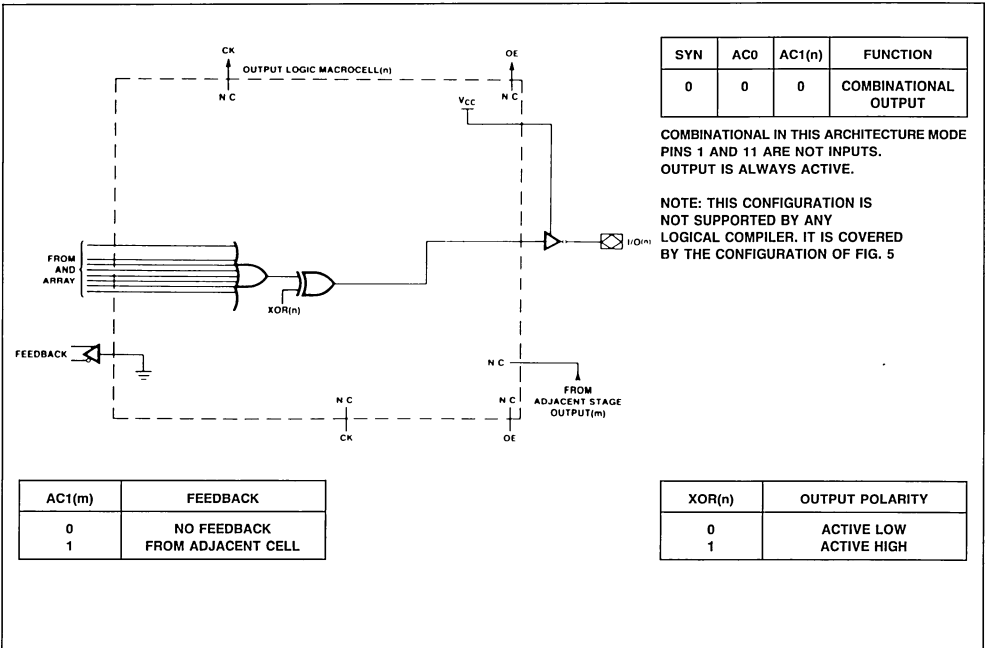


Fig. 8 - Dedicated Combinational Output

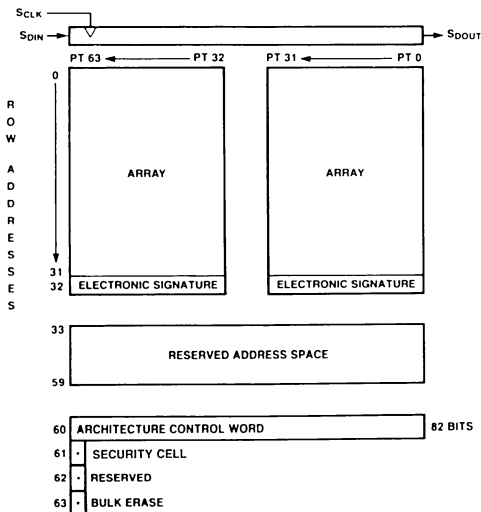


ROW ADDRESS MAP DESCRIPTION

Figure 9 shows a block diagram of the row address map. There are a total of 36 unique row addresses available to the user when programming the GAL16V8 devices. Row addresses 0-31 each contain 64 bits of input term data. This is the user array where the custom logic pattern is programmed. Row 32 is the Electronic Signature Word. It has 64 bits available for any user defined purpose. Row 33-59 are reserved by the manufacturer and are not available to users.

Row 60 contains the architecture and output polarity information. The 82 bits within this word are programmed to configure the device for a specific application. Row 61 contains a one bit security cell that when programmed prevents further programming verification of the array. Row 63 is the row that is addressed to perform a bulk erase of the device, resetting it back to a virgin state. Each of these functions is described in the following sections.

Figure 9. GAL16V8 Row Address Map Block Diagram



ELECTRONIC SIGNATURE WORD DESCRIPTION

An Electronic Signature Word is provided with every GAL16V8 device. It resides at row address 32 and contains 64 bits of reprogrammable memory that can contain user-defined data. Some uses include user ID codes, revision numbers, or inventory control. This signature data is always available to the user independent of the state of the security Cell.

ARCHITECTURE CONTROL WORD

All of the various configurations of the GAL16V8 devices are controlled by programming cells within the 82 bit Architecture Control Word that resides at row 60. The location of specific bits within the Architecture Control Word is shown in the control word diagram in Figure 10. The function of the SYN, AC0 and AC1(n) bits have been explained in the OUTPUT LOGIC MACROCELL description. The eight polarity bits determine each output's polarity individually by selectively correct logic. The numbers below the XOR(n) and AC1(n) bits in the architecture control word diagram shows the output device pin number that the polarity bits control.

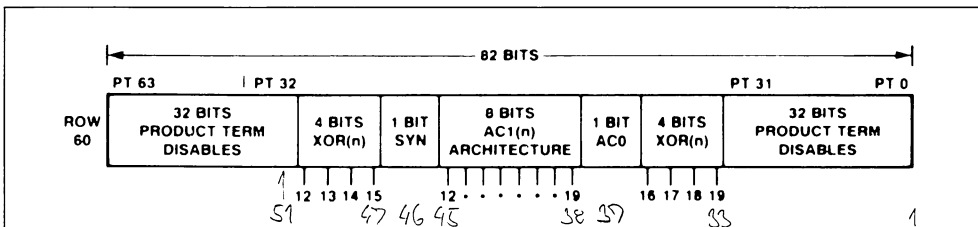
SECURITY CELL

Row address 61 contains the Security Cell (one bit). The Security Cell is provided on all GAL16V8 devices as a deterrent to unauthorized copying of the array configuration patterns. Once programmed, the circuitry enabling array access is disabled, preventing further programming or verification of the array (rows 0-31). The cell can be erased only in conjunction with the array during a bulk erase cycle, so the original configuration can never be examined once this cell is programmed. Signature data is always available to the user.

BULK ERASE MODE

By addressing row 63 during a programming cycle, a clear function performs a bulk erase of the array and the Architecture Control Word. In addition, the Electronic Signature Word and the Security Cell are erased. This mode resets a previously configured device back to its virgin state.

Figure 10. GAL16V8 Architecture Control Word Diagram

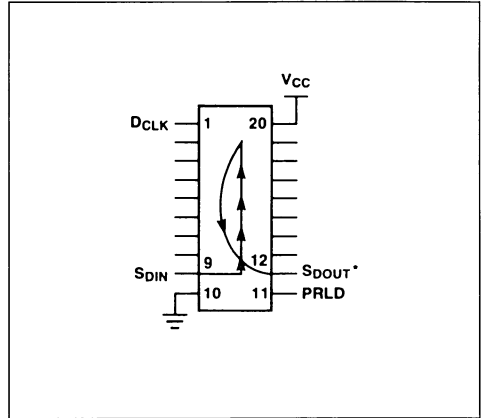


OUTPUT REGISTER PRELOAD

When testing state machine designs, all possible states and state transitions must be verified in the design, not just those required in the normal machine operations. This is because in system operation, certain events occur that may throw the logic into an illegal state (power-up, line voltage glitches, brown-outs, etc.). To test a design for proper treatment of these conditions, a way must be provided to break the feedback paths, and force any desired (i.e. illegal) state into a registers. Then the machine can be sequenced and the outputs tested for the correct next state conditions.

The GAL16V8 device includes circuitry that allows each registered output to be synchronously set either high or low. Thus, any present state condition can be forced for test sequencing. Figure 11 shows the pin functions necessary to preload the register. The register preload timing and pin voltage levels necessary to perform the function are shown below (see fig. 12). This test mode is entered by raising PRLD to V_{IES} , which enables the serial data in (S_{DIN}) buffer and the serial data out (S_{DOUT}) buffer. Data is then serially shifted in the registers on each rising edge of the clock, DLCK. Only the macrocells with registered output configurations are loaded. If only 3 outputs have registers, then only 3 bits need be shifted in. The registers are loaded from the bottom up as shown in figure 11.

Figure 11. Output Register Preload Pinout



LATCH-UP PROTECTION

GAL devices are designed with on board charge pump to negatively bias the substrate. The negative bias is of sufficient magnitude to prevent input undershoots from causing the circuitry to latch. Additionally, outputs are designed with n-channel pullups instead of the traditional p-channel pullups to eliminate any possibility of SCR induced latching.

Figure 12. Register Preload Waveforms

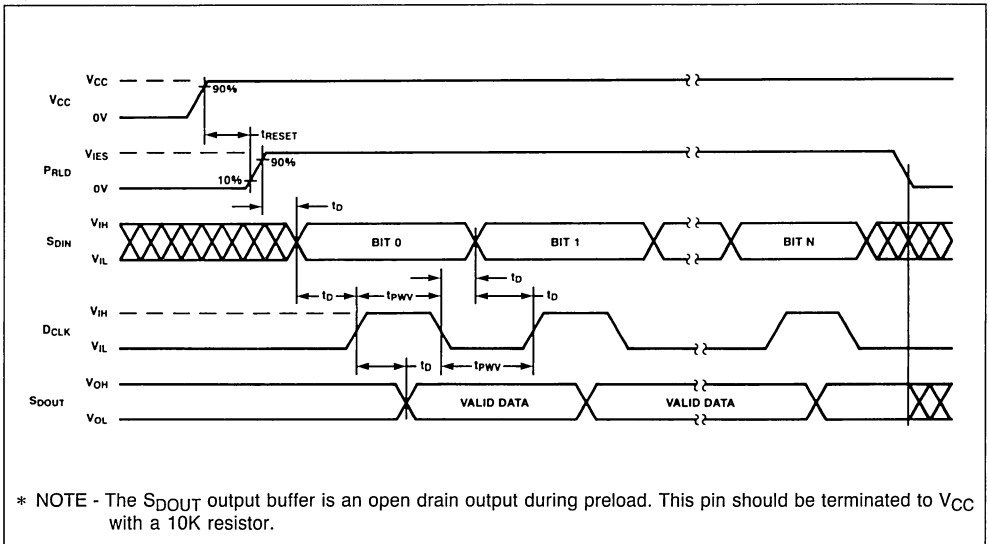
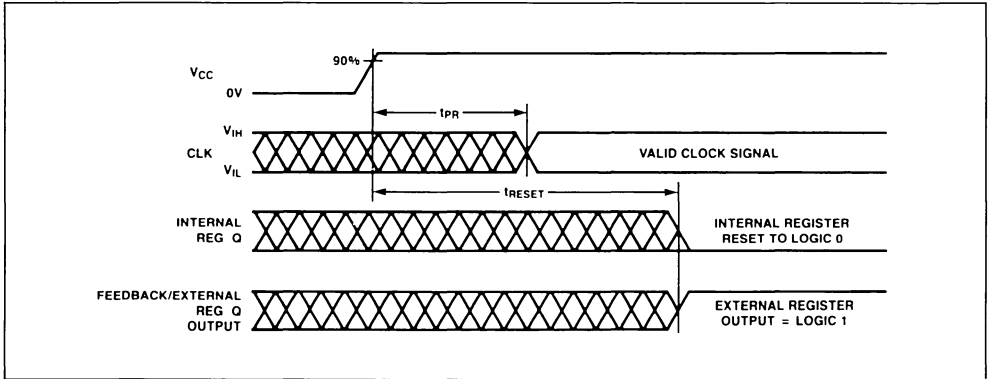


Figure 13. Power-up Reset



LATCH-UP PROTECTION (Continued)

Circuitry within the GAL16V8 provides a reset signal to all registers during power-up (see fig. 13). All internal registers will have their Q outputs set low after a specified time (t_{RESET}). As a result, the state on the registered output pins (if they are enabled through OE) will always be high on power-up, regardless of the programmed polarity of the output pins. This feature can greatly simplify state machine design by providing a known state on power-up.

The timing diagram for power-up is shown above. Because of asynchronous nature of system power-up, some conditions must be met to guarantee a valid power-up reset of the GAL16V8. First, the V_{CC} rise must be monotonic. Second, the clock input must become a proper TTL level within the specified time (t_{PR}). The registers will reset within a maximum of t_{RESET} time. As in normal system operation, avoid clocking the device until all input and feedback path setup times have been met.

FIELD SUPPORT TOOLS

Although it is possible to program GAL devices manually, SGS-THOMSON strongly recommends the use of approved programming hardware and software. Programming on unapproved equipment will generally void all guarantees.

PROGRAMMER/DEVELOPMENT SYSTEMS

VENDOR	SYSTEM	REVISION
DATA I/O	Model 29B	V04
	Adapter 303A-011A	V06
	Unisite	V1.7
STAG	Model 60A	V12
	Adapter 360-001A	
STAG	Model ZL30A	30A28
Programmable Logic Tech.	Logic Lab	V2.1 (◇)
Qwerty	QPR-1000 +	2.0

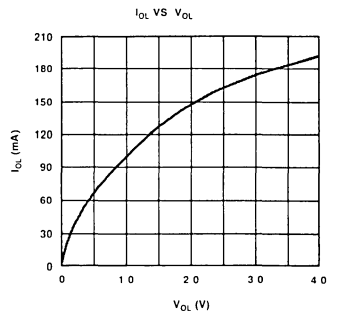
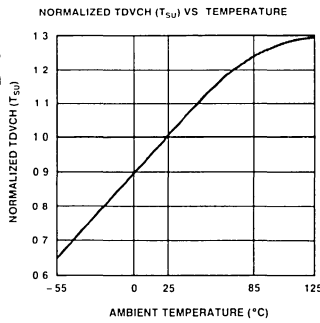
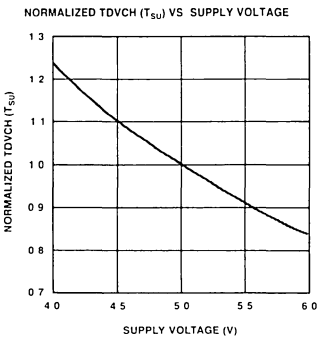
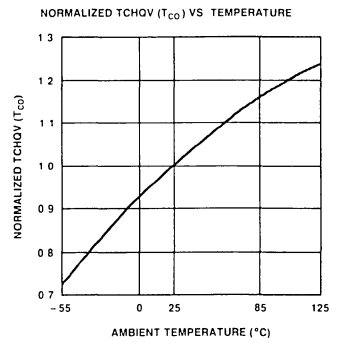
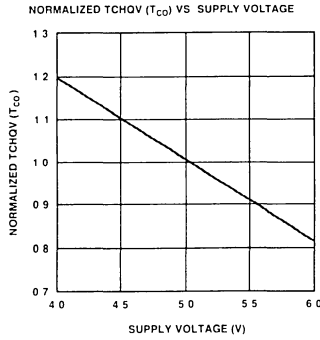
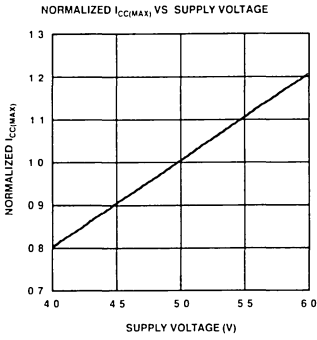
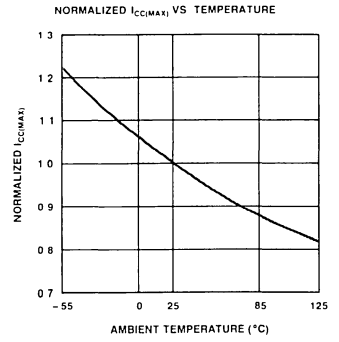
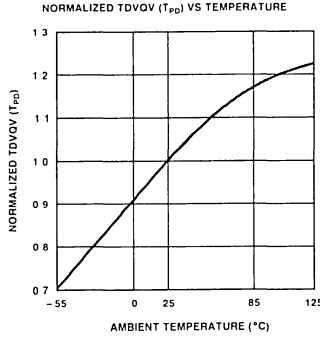
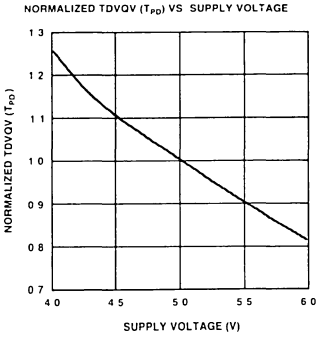
(◇) In conjunction with IBM PC (or compatible) using FastMap Plus* V2.1 software.

SOFTWARE DEVELOPMENT TOOLS†

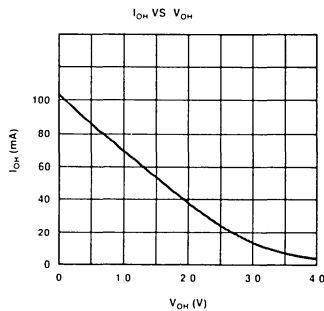
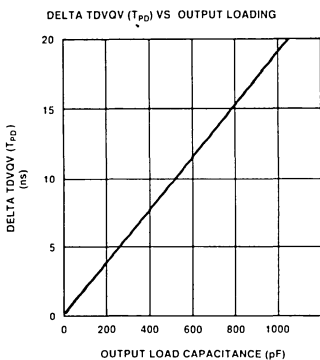
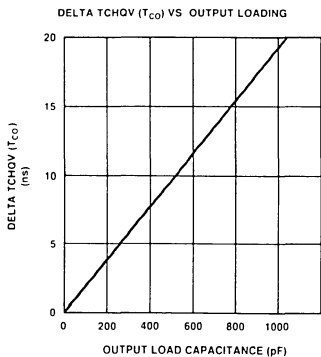
PACKAGE	VENDOR	REVISION
CUPL*	Personal Cad Systems	V2.15
ABEL*	Data I/O	V3.0
PLDtest*	Data I/O	V1.0 (†)
DASH-ABEL	Data I/O	V1.0
PALASM*	Monolithic Memories Inc.	(†)
LC9000*	Programmable Logic Tec.	V1.5
PLAQ*	Qwerty Inc.	V1.0

(†) When emulating PAL devices any revision of the software can be used to create the PAL JEDEC file. Some programming hardware will automatically configure the GAL architecture.

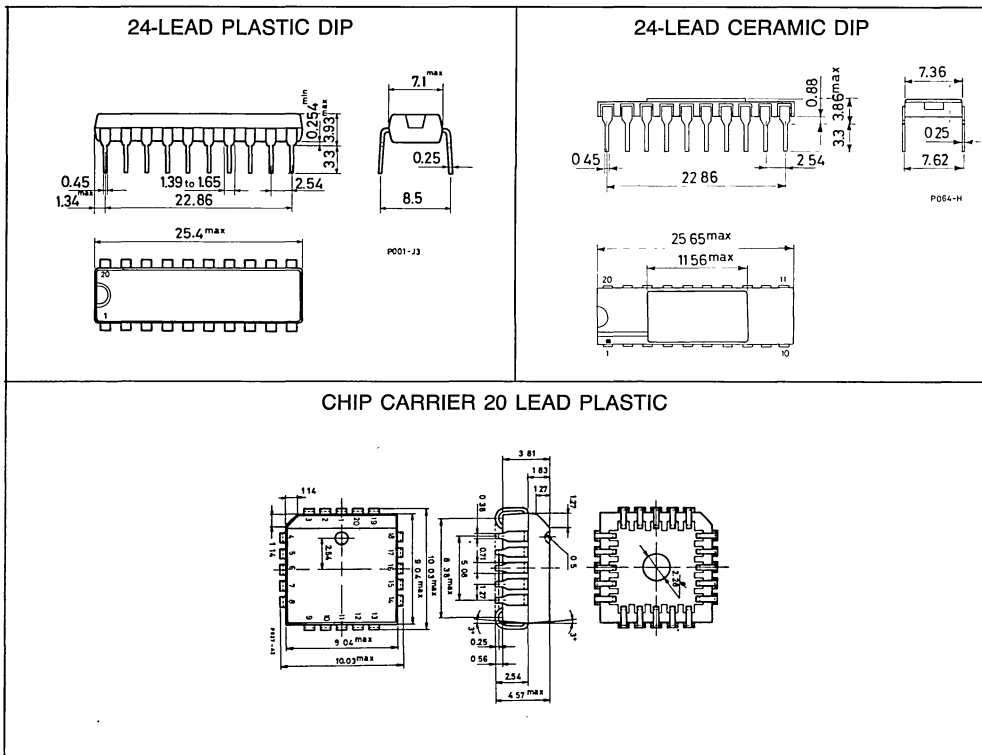
GAL16V8 SPECIFICATION (Typical values)



GAL16V8 SPECIFICATION (Continued)



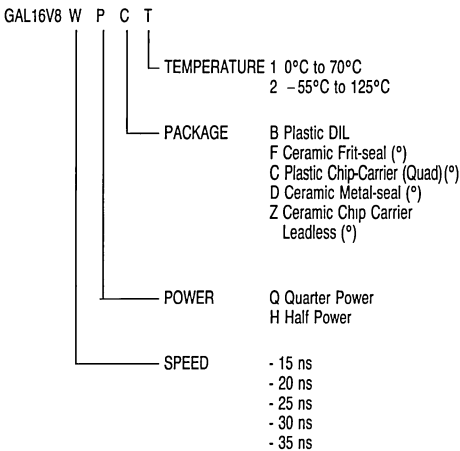
PACKAGE INFORMATION



GAL16V8 SPECIFICATION (Continued)

ORDERING INFORMATION

The SGS-THOMSON GAL devices are available in a variety of packages and two temperature ranges. General ordering code is reported below.



EXAMPLE: Order Code for a GAL16V8, 20 ns speed and half power in plastic DIL is: **GAL16V8-20HB1**

(°) Please contact SGS-THOMSON Sales Office

- Burn-in (168h, 125°C) also available for these products (add suffix "X").
- Military temperature range is available in ceramic packages only.

SPEED/POWER CROSS-REFERENCE GUIDE

SPEED	POWER	GAL DEVICE	BIPOLAR PAL DEVICE
15ns	45mA	- 15Q	—
15ns	90mA	- 15H	—
15ns	180mA	use - 15H	B
(♦) 20ns	50mA	- 20Q	—
(♦) 20ns	90mA	- 20H	—
(♦) 20ns	210mA	use - 20H	B MIL
25ns	45mA	- 25Q	—
25ns	90mA	- 25H	B-2
25ns	180mA	use - 25H	A
(♦) 30ns	50mA	- 30Q	—
(♦) 30ns	90mA	- 30H	B-2 MIL
(♦) 30ns	210mA	use - 30H	A MIL
35ns	45mA	- 35Q	B-4
35ns	90mA	use - 35Q	A-2
35ns	180mA	use - 35Q	STD
(♦) 40ns	50mA	use - 30Q	B-4 MIL
(♦) 40ns	90mA	use - 30H	A-2 MIL
(♦) 40ns	210mA	use - 30H	STD MIL

(♦) MILITARY TEMPERATURE RANGE

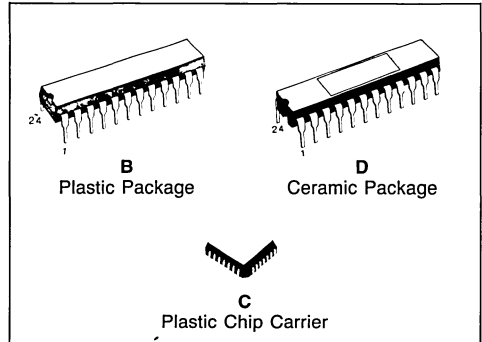
E²CMOS PROGRAMMABLE LOGIC DEVICE

- **ELECTRICALLY ERASABLE CELL TECHNOLOGY**
 - Reconfigurable Logic
 - Reprogrammable Cells
 - Guaranteed 100% Yields
- **HIGH PERFORMANCE E²CMOS TECHNOLOGY**
 - Low Power : 90/70mA Max Active/Standby (Half Power)
 - 45/35mA Max Active/Standby (Quarter Power)
 - High Speed: 15 to 25ns Access Max (Half Power)
 - 20 to 35ns Access Max (Quarter Power)
- **EIGHT OUTPUT LOGIC MACROCELLS**
 - Maximum Flexibility for Complex Logic Designs
 - Also Emulates 24-pin PAL* Devices with Full Function/Fuse Map/Parametric Compatibility
- **PRELOAD AND POWER-ON RESET OF ALL REGISTERS**
 - 100% Functional Testability
- **HIGH SPEED PROGRAMMING ALGORITHM**
- **SECURITY CELL PREVENTS COPYING LOGIC**
- **DATA RETENTION EXCEEDS 20 YEARS**

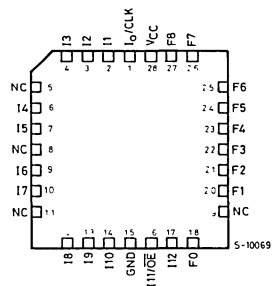
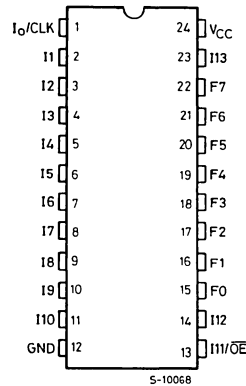
DESCRIPTION

The GAL20V8 E²CMOS device combines a high performance CMOS process with electrically erasable floating gate technology. This programmable memory technology applied to array logic provides designers with reconfigurable logic and bipolar performance at significantly reduced power levels. The 24-pin GAL20V8 features 8 programmable Output Logic Macrocells (OLMCs) allowing each output to be configured by the user. Additionally, the GAL20V8 is capable of emulating, in a functional/fuse map/parametric compatible device, all 24 pin PAL device architectures (excluding XOR devices).

Programming is accomplished using readily available hardware and software tools. SGS-THOMSON guarantees 100 erase/write cycles and data retention exceeds 20 years. Unique test circuitry and reprogrammable cells allows complete AC,DC cell and functionality testing during manufacture. Therefore, SGS-THOMSON guarantees 100% field programmability and functionality of the GAL devices. In addition, electronic signature is available to provide positive device ID. A security circuit is built-in, providing proprietary designs with copy protection.



PIN CONNECTIONS (top view)



Contact factory for JEDEC Pin Configuration availability on PLCC Package

GAL20V8

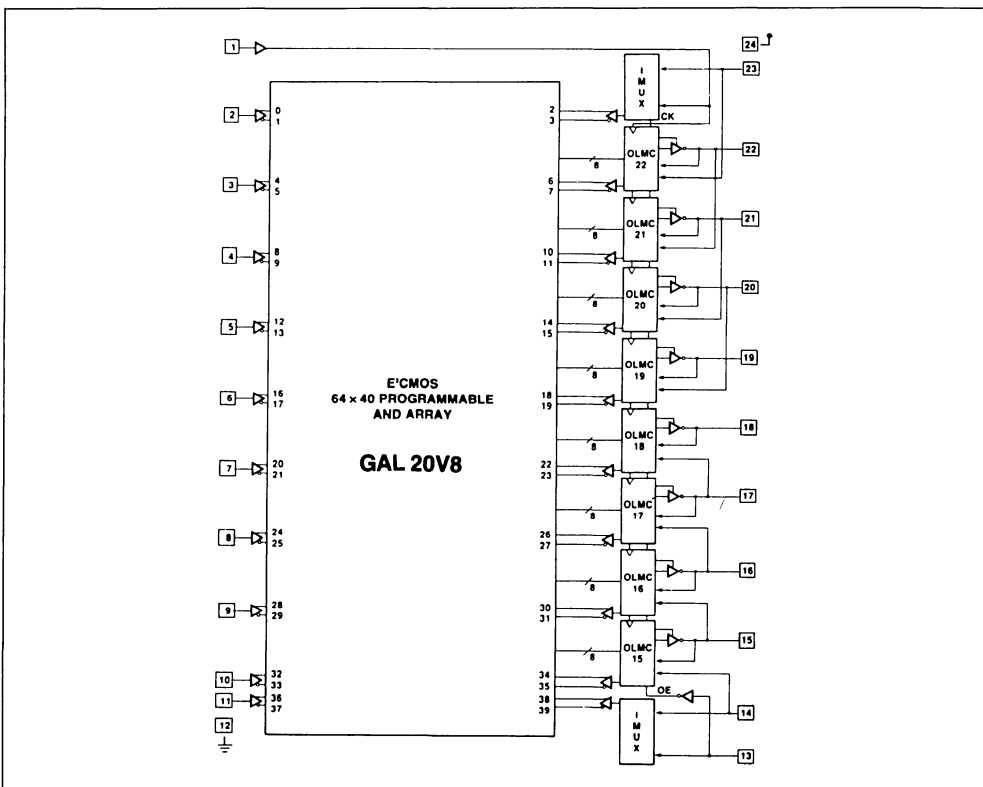
PIN NAMES

I0-I19	INPUT
CK	CLOCK INPUT
B0-B5	BIDIRECTIONAL
F0-F7	OUTPUT
\overline{OE}	OUTPUT ENABLE
V _{CC}	POWER (+5V)
GND	GROUND

GAL20V8 EMULATING PAL DEVICES

I ₀ /CK	1	24	V _{CC}								
11			I13	I15	I17	I19	I12	I12	I12	I13	
12			F7	I14	I16	I18	F7	B1	B3	F1	
13			F6	F5	I15	I17	F6	F5	B2	B5	
14			F5	F4	F3	I16	F5	F4	F3	B4	
15			F4	F3	F2	F1	F4	F3	F2	B3	
16			F3	F2	F1	F0	F3	F2	F1	B2	
17			F2	F1	F0	I15	F2	F1	F0	B1	
18			F1	F0	I14	I14	F1	F0	B1	B0	
19			F0	I13	I13	I13	F0	B0	B0	F0	
I10			I12	I12	I12	I11	I11	I11	I11	I12	
GND	12	13	I11	I11	I11		\overline{OE}	\overline{OE}	\overline{OE}	I11	
				14L8	16L6	18L4	20L2	20R8	20R6	20R4	20L8
				14H8	16H6	18H4	20H2	20RP8	20RP6	20RP4	20H8
				14P8	16P6	18P4	20P2				20P8
GAL20V8											

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	- 0.5 to + 7	V
V_I	Input Voltage Applied	- 2.5 to $V_{CC} + 1.0$	V
V_O	Off-State Output Voltage Applied	- 2.5 to $V_{CC} + 1.0$	V
T_{stg}	Storage Temperature	- 65 to 125	°C

Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specification).

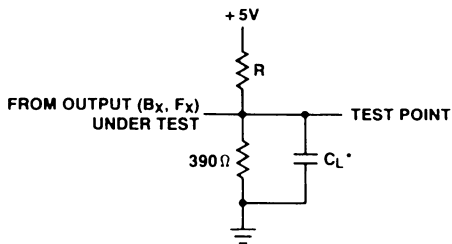
OPERATING RANGE

Symbol	Parameter	Temperature Range						Unit
		Military			Commercial			
		Min.	Typ.	Max.	Min.	Typ.	Max.	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
T_A	Ambient Temperature				0		70	°C
T_C	Case Temperature	- 55		125				°C

SWITCHING TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10%-90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See figure

3-state levels are measured 0.5V from steady-state active level.



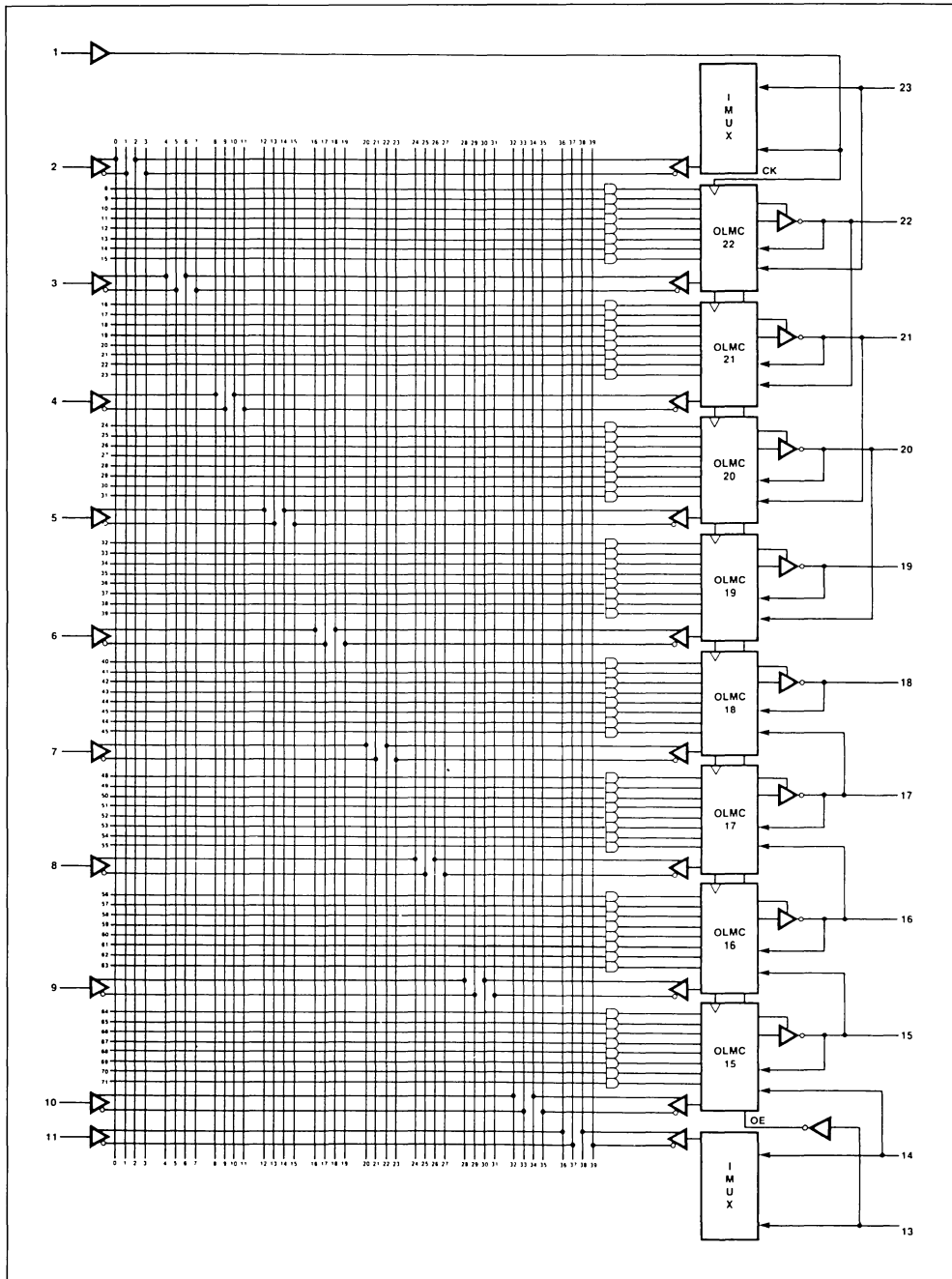
(°) C_L INCLUDES JIG AND PROBE TOTAL CAPACITANCE

CAPACITANCE ($T_A = 25^\circ\text{C}$, $f = 1.0\text{MHz}$, $V_{CC} = 5\text{V}$)

Symbol	Parameter	Test Conditions	Maximum*	Units
C_I	Input Capacitance	$V_I = 2\text{V}$	12	pF
C_F	Output Capacitance	$V_F = 2\text{V}$	15	pF
C_B	Bidirectional Pin Cap	$V_B = 2\text{V}$	15	pF

* Guaranteed but not 100% tested

LOGIC DIAGRAM



ELECTRICAL CHARACTERISTICS OVER OPERATING CONDITIONS
PRELIMINARY - Quarter Power GAL20V8

Symbol	Parameter	Test Conditions	Temp. Range	Min.	Max.	Unit
I_{IH}, I_{IL}	Input Leakage Current	$GND \leq V_{IN} \leq V_{CC} \text{ MAX}$		—	± 10	μA
I_{BZH}, I_{BZL}	Bidirectional Input Leakage Current	$GND \leq V_{IN} \leq V_{CC} \text{ MAX}$		—	± 10	μA
I_{FZL}, I_{FZH}	Output Pin Leakage Current	$GND \leq V_{IN} \leq V_{CC} \text{ MAX}$		—	± 10	μA
I_{CC}	Operating Power Supply Current	$f = 15\text{MHz}$ $V_{CC} = V_{CC} \text{ MAX}$	COM'L	—	45	mA
			MIL		50	mA
$I_{OS}(\text{°})$	Output Short Circuit Current	$V_{CC} = 5.0\text{V}$ $V_{OUT} = GND$		-30	-130	mA
I_{SB}	Standby Power Supply Current	$V_{CC} = V_{CC} \text{ MAX}$	COM'L	—	35	mA
			MIL		45	mA
V_{OL}	Output Low Voltage $V_{CC} = V_{CC} \text{ MIN}$	$I_{OL} = 24\text{mA}$	COM'L		0.5	V
		$I_{OL} = 12\text{mA}$	MIL		0.5	V
V_{OH}	Output High Voltage $V_{CC} = V_{CC} \text{ MIN}$	$I_{OH} = 3.2\text{mA}$	COM'L	2.4	—	V
		$I_{OH} = -2.0\text{mA}$	MIL	2.4	—	V
V_{IH}	Input High Voltage			2.0	$V_{CC} + 1$	V
V_{IL}	Input Low Voltage			—	0.8	V

(°) One output at a time for a maximum duration of one second.

Half Power GAL20V8

Symbol	Parameter	Test Conditions	Temp. Range	Min.	Max.	Unit
I_{IH}, I_{IL}	Input Leakage Current	$GND \leq V_{IN} \leq V_{CC} \text{ MAX}$		—	± 10	μA
I_{BZH}, I_{BZL}	Bidirectional Pin Leakage Current	$GND \leq V_{IN} \leq V_{CC} \text{ MAX}$		—	± 10	μA
I_{FZL}, I_{FZH}	Output Pin Leakage Current	$GND \leq V_{IN} \leq V_{CC} \text{ MAX}$		—	± 10	μA
I_{CC}	Operating Power Supply Current	$f = 15\text{MHz}$ $V_{CC} = V_{CC} \text{ MAX}$	COM'L	—	90	mA
			MIL		90	mA
$I_{OS}(\text{°})$	Output Short Circuit Current	$V_{CC} = 5.0\text{V}$ $V_{OUT} = GND$		-30	-130	mA
I_{SB}	Standby Power Supply Current	$V_{CC} = V_{CC} \text{ MAX}$	COM'L	—	70	mA
			MIL	—	70	mA

(°) One output at a time for a maximum duration of one second.

ELECTRICAL CHARACTERISTICS OVER OPERATING CONDITIONS (Continued)
PRELIMINARY - Quarter Power GAL20V8

Symbol	Parameter	Test Conditions	Temp. Range	Min.	Max.	Unit
V _{OL}	Output Low Voltage V _{CC} = V _{CC} MIN	I _{OL} = 24mA	COM'L		0.5	V
		I _{OL} = 12mA	MIL		0.5	V
V _{OH}	Output High Voltage V _{CC} = V _{CC} MIN	I _{OH} = -3.2mA	COM'L	2.4	—	V
		I _{OH} = -2.0mA	MIL	2.4	—	V
V _{IH}	Input High Voltage			2.0	V _{CC} + 1	V
V _{IL}	Input Low Voltage			—	0.8	V

SWITCHING CHARACTERISTICS OVER OPERATING CONDITIONS
PRELIMINARY - Quarter Power GAL20V8

Symbol	Parameter	TEMPERATURE RANGE								Units	Test Conditions(°)			
		0°C to 70°C									-55 to 125°C		R (Ω)	C _L (pF)
		20V8-20		20V8-25		20V8-35		20V8-30						
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.					
T _{DVQV1}	Delay from Input to active output (°°°)	—	20	—	25	—	35	—	30	ns	200	50		
T _{DVQV2}	Product Term Enable Access Time to Active Output	—	20	—	25	—	35	—	30	ns	Active High R = ∞ Active Low R = 200	50		
T _{DVQZ} (°°)	Product Term Enable to Outputs Off	—	20	—	25	—	35	—	30	ns	From V _{OH} R = ∞ From V _{OL} R = 200	5		
T _{GHQZ} (°°)	\overline{OE} Output Enable High to Output Off	—	18	—	20	—	25	—	25	ns	From V _{OH} R = ∞ From V _{OL} R = 200	5		
T _{GLQV}	\overline{OE} Output Enable Access Time	—	18	—	20	—	25	—	25	ns	Active High R = ∞ Active Low R = 200	50		
T _{CHQV}	Clock High to Output Valid Access Time	—	15	—	15	—	25	—	25	ns	200	50		
T _{DVCH}	Input or Feedback Data Minimum Setup Time	—	15	—	20	—	30	—	25	ns	—	—		
T _{CHDX}	Input or Feedback Data Minimum Hold Time	—	0	—	0	—	0	—	0	ns	—	—		
T _{CHCH}	Minimum Clock Period (T _{DVCH} + T _{CHQV})	—	30	—	35	—	55	—	45	ns				
T _{CHCL}	Minimum Clock Width HIGH	—	12	—	15	—	20	—	15	ns	—	—		
T _{CLCH}	Minimum Clock Width LOW	—	12	—	15	—	20	—	15	ns				
f _{MAX}	Maximum Frequency	SYNCH.	33.3	—	28.5	—	18.1	—	22.2	—	MHz	200	50	
		ASYNCH.	50.0	—	40.0	—	28.5	—	33.3	—				

(°) Refer also to switching test conditions.

(°°) 3-state levels are measured 0.5V from steady-state active level.

(°°°) XOR (n) = 1 (active high)

SWITCHING CHARACTERISTICS OVER OPERATING CONDITIONS

Half Power GAL20V8

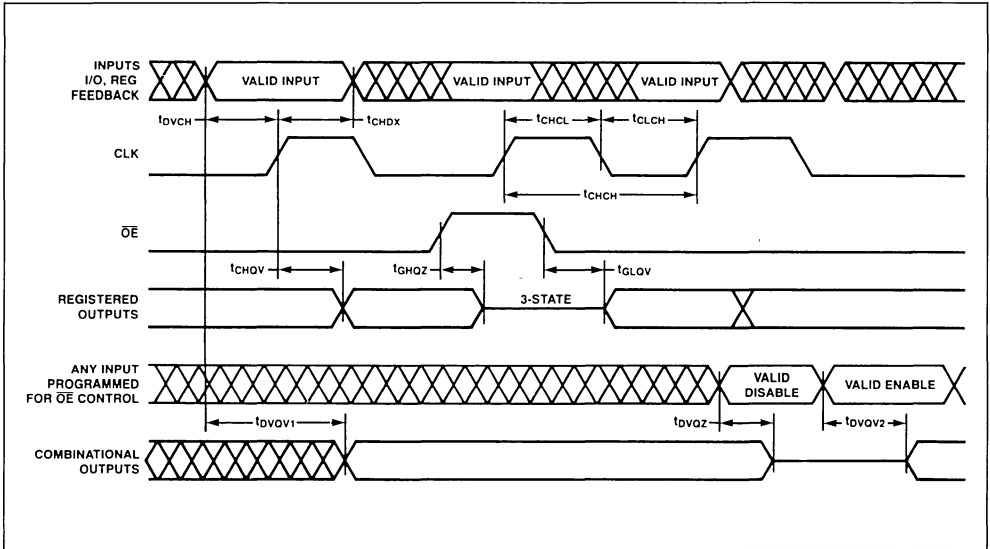
Symbol	Parameter	TEMPERATURE RANGE										Units	Test Conditions(°)	
		0°C to 70°C						-55 to 125°C					R (Ω)	C _L (pF)
		20V8-15		20V8-20		20V8-25		20V8-20		20V8-30				
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.			
T _{DVQV1}	Delay from Input to active output (°°°)	—	15	—	20	—	25	—	20	—	30	ns	200	50
T _{DVQV2}	Product Term Enable Access Time to Active Output	—	15	—	20	—	25	—	20	—	30	ns	Active High R=∞ Active Low R=200	50
T _{DVQZ} (°°)	Product Term Enable to Outputs Off	—	15	—	20	—	25	—	20	—	30	ns	From V _{OH} R=∞ From V _{OL} R=200	5
T _{GHQZ} (°°)	\overline{OE} Output Enable High to Output Off	—	15	—	18	—	20	—	18	—	25	ns	From V _{OH} R=∞ From V _{OL} R=200	5
T _{GLQV}	\overline{OE} Output Enable Access Time	—	15	—	18	—	20	—	18	—	25	ns	Active High R=∞ Active Low R=200	50
T _{CHQV}	Clock High to Output Valid Access Time	—	12	—	15	—	15	—	15	—	20	ns	200	50
T _{DVCH}	Input or Feedback Data Minimum Setup Time	—	12	—	15	—	20	—	15	—	25	ns	—	—
T _{CHDX}	Input or Feedback Data Minimum Hold Time	—	0	—	0	—	0	—	0	—	0	ns	—	—
T _{CHCH}	Minimum Clock Period (T _{DVCH} + T _{CHQV})	—	24	—	30	—	35	—	30	—	45	ns	—	—
T _{CHCL}	Minimum Clock Width HIGH	—	10	—	12	—	15	—	12	—	15	ns	—	—
T _{CLCH}	Minimum Clock Width LOW	—	10	—	12	—	15	—	12	—	15	ns	—	—
f _{MAX}	Maximum Frequency	SYNCH.	41.6	—	33.3	—	28.5	—	33.3	—	22.2	MHz	200	50
		ASYNCH.	66.6	—	50.0	—	40.0	—	50.0	—	33.3			

(°) Refer also to switching test conditions.

(°°) 3-state levels are measured 0.5V from steady-state active level.

(°°°) XOR (n) = 1 (active high)

SWITCHING WAVEFORMS



OUTPUT LOGIC MACROCELL (OLMC)

The following discussion pertains to configuring the Output Logic Macrocell. It should be noted that actual implementation is accomplished by development software/hardware and is completely transparent to the user.

The outputs of the AND array are fed into an OLMC, where each output can be individually set to active high or active low, with either combinational (asynchronous) or registered (synchronous) configurations. A common output enable can be connected to all outputs, or separate inputs or product terms can be used to provide individual output enable controls. The Output Logic Macrocell provides the designer with maximal output flexibility in matching signal requirements, thus providing more functions than possible with existing 24-pin PAL devices.

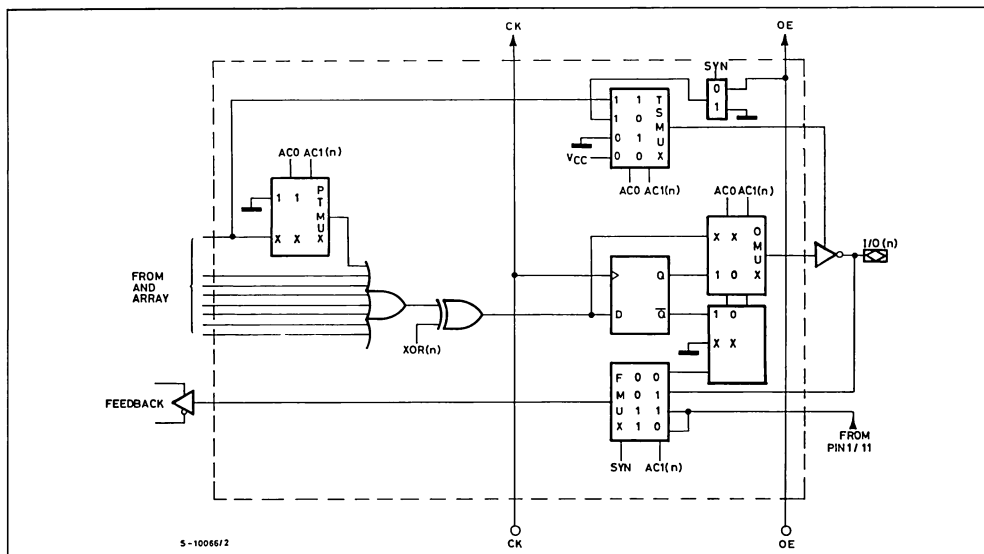
The various configurations of the Output Logic Macrocell are controlled by programming certain cells (SYN, AC0, AC1(n) and the XOR(n) polarity bits)

within the 82-bit Architecture Control Word. The SYN bit determines whether a device will have registered output capability or purely combinational outputs. It also replaces the AC0 bit in the two outermost macrocells, OLMC(15) and OLMC(22). When first setting up the device architecture, this is the first bit to choose.

Architecture Control bit AC0 and the eight AC1(n) bits direct the outputs to be wired always on, always off (as an input), have a common OE term (pin 13), or to be three-state controlled separately from a product term. The Architecture Control bits also determine the source of the array feedback term through the FMUX, and select either combinational or registered outputs.

The eight macrocell configurations are shown in each of the macrocell equivalent diagrams, (see also table 1 and 2). In all cases, the eight XOR(n) bits individually determine each output's polarity. The truth table associated with each diagram shows the bit values of the SYN, AC0, and AC1(n) that set the macrocell to the configuration shown.

GAL20V8 Output Logic Macrocell: Pin 15 and 22



GAL20V8 Output Logic Macrocell: Pin 16 to 21

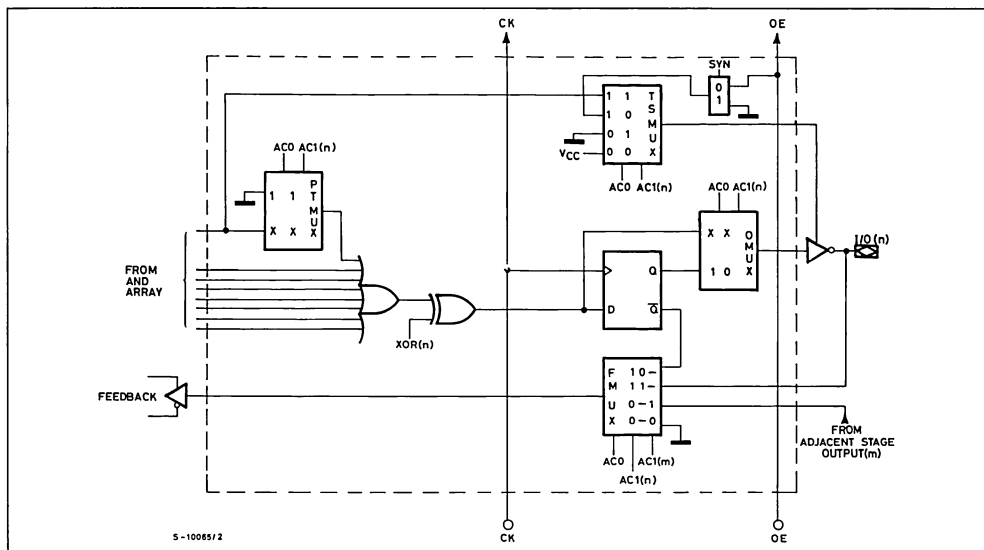


TABLE 1 - ARCHITECTURE OLMC (Pin 15 and 22)

SYN	AC0	AC1(n)	P T MUX	T S MUX	O MUX	F MUX
0	0	0	8 PT	Active OUT	Comb. OUT	No Feedback
0	0	1	Don't Care	3-State OUT	Don't Care	Input Mode
0	1	0	8 PT	Common OE	Regis. OUT	Direct Registered Feedback
0	1	1	7 PT	Single OE (PT)	Comb. OUT	Direct Comb. Feedback or Input
1	0	0	8 PT	Active OUT	Comb. OUT	Pin 1(22) / 13(15)
1	0	1	Don't Care	3-State OUT	Don't Care	Pin 1(22) / 13(15)
1	1	0	Don't Care	3-State OUT	Don't Care	Pin 1(22) / 13(15)
1	1	1	7 PT	Single OE (PT)	Comb. OUT	Pin 1(22) / 13(15)

} Pins 1 and 13
Become Data Inputs

XOR (n)	Output Polarity
0	ACTIVE LOW
1	ACTIVE HIGH

TABLE 2 - ARCHITECTURE OLMC (Pin 16 to 21)

SYN	AC0	AC1(n)	AC1(m)	P T MUX	T S MUX	O MUX	F MUX
0	0	0	0	8 PT	Active OUT	Comb. OUT	No Feedback
0	0	0	1	8 PT	Active OUT	Comb. OUT	Adjacent Macrocell
0	0	1	0	Don't Care	3-State OUT	Don't Care	No Feedback
0	0	1	1	Don't Care	3-State OUT	Don't Care	Adjacent Macrocell
0	1	0	0	8 PT	Common OE	Regis. OUT	Direct Registered Feedback
0	1	0	1	8 PT	Common OE	Regis. OUT	Direct Registered Feedback
0	1	1	0	7 PT	Single OE (PT)	Comb. OUT	Direct Comb. Feedback or Input
0	1	1	1	7 PT	Single OE (PT)	Comb. OUT	Direct Comb. Feedback or Input
1	0	0	0	8 PT	Active OUT	Comb. OUT	No Feedback
1	0	0	1	8 PT	Active OUT	Comb. OUT	Adjacent Macrocell
1	0	1	0	Don't Care	3-State OUT	Don't Care	No Feedback
1	0	1	1	Don't Care	3-State OUT	Don't Care	Adjacent Macrocell
1	1	0	0	Don't Care	3-State OUT	Don't Care	No Feedback
1	1	0	1	Don't Care	3-State OUT	Don't Care	No Feedback
1	1	1	0	7 PT	Single OE (PT)	Comb. OUT	Direct Comb. Feedback or Input
1	1	1	1	7 PT	Single OE (PT)	Comb. OUT	Direct Comb. Feedback or Input

XOR (n)	OUTPUT POLARITY
0	ACTIVE LOW
1	ACTIVE HIGH

Fig. 1 - Combinational Output

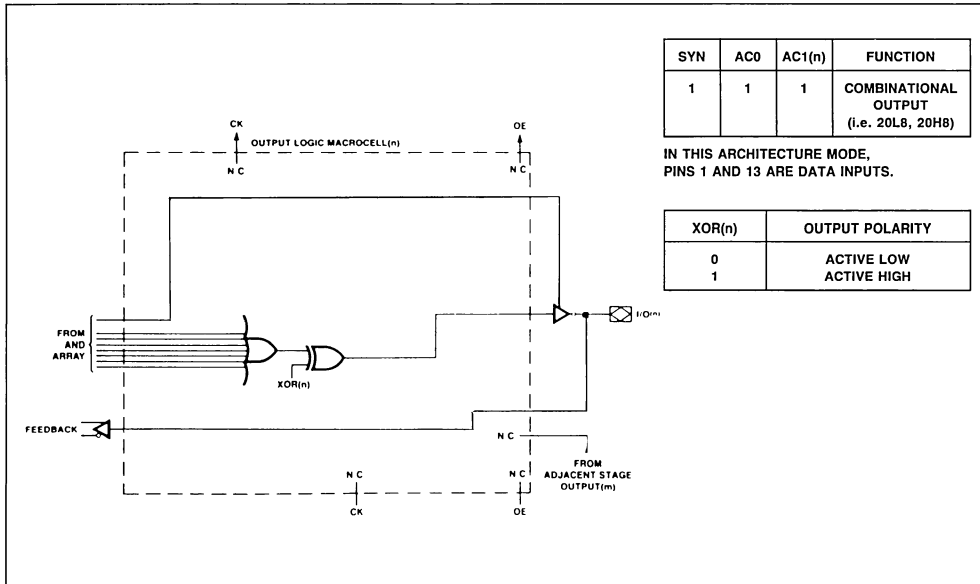


Fig. 2 - Disabled Output

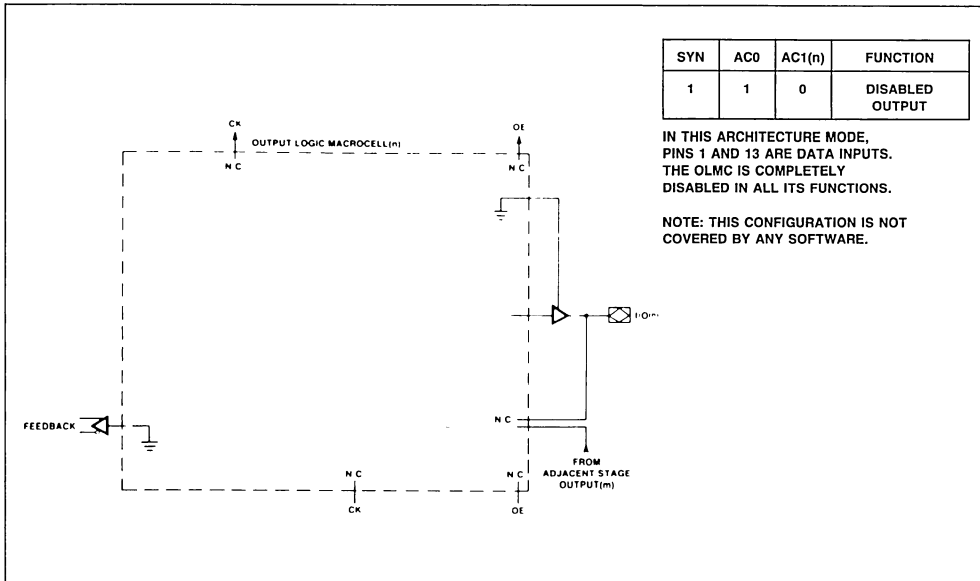


Fig. 3 - Combinational Output in a Registered Device

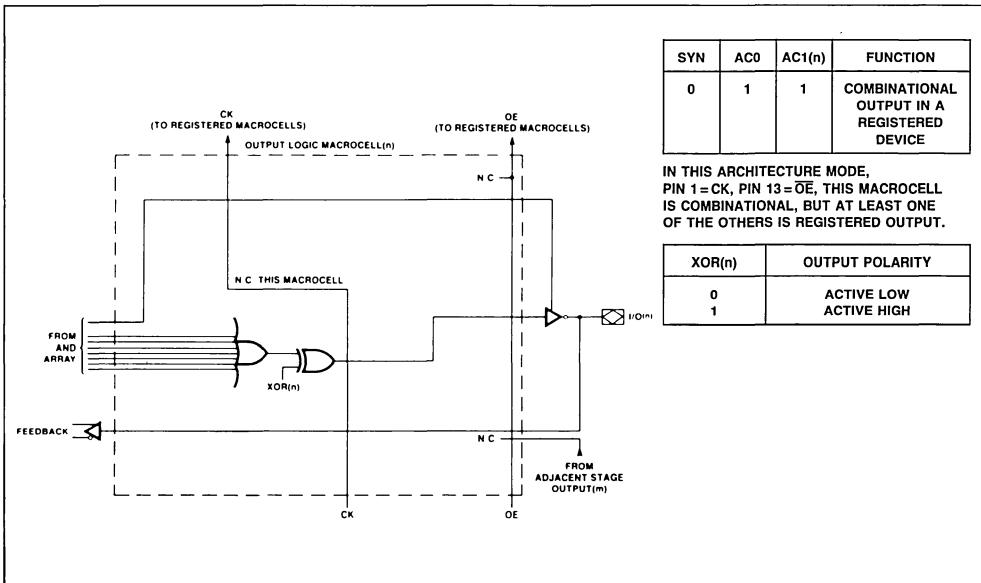


Fig. 4 - Registered Active High or Low Output

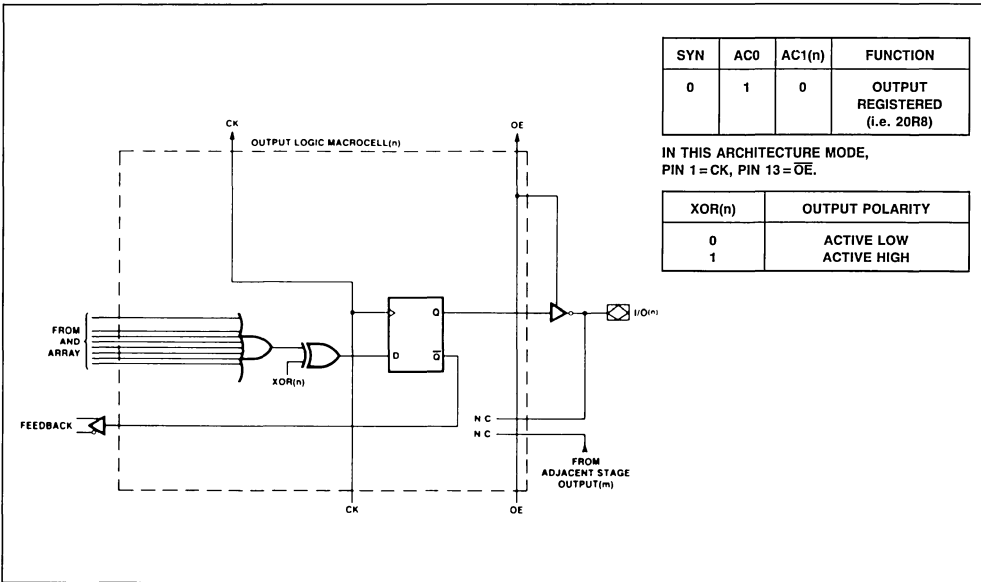


Fig. 5 - Dedicated Combinational Output

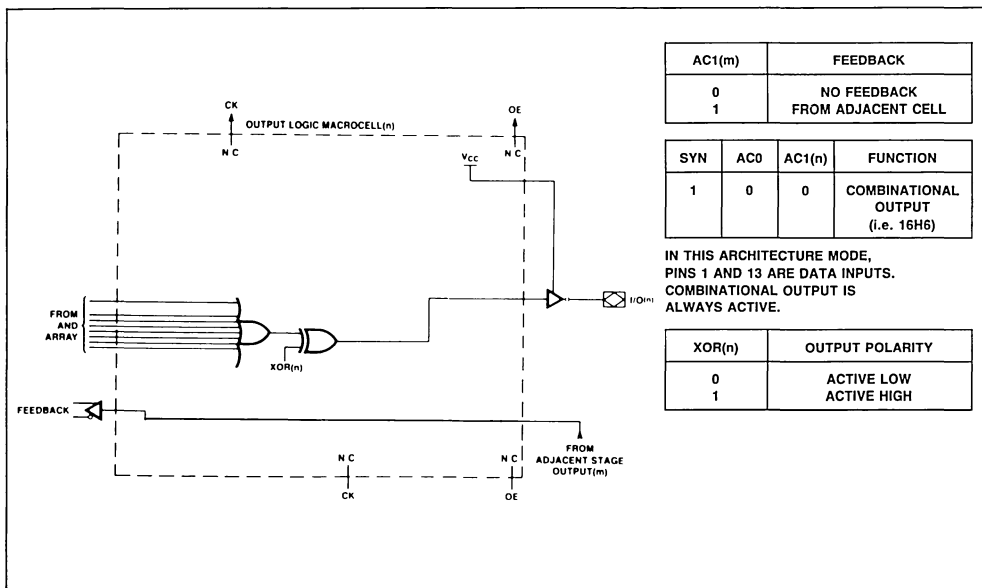


Fig. 6 - Dedicated Input Mode

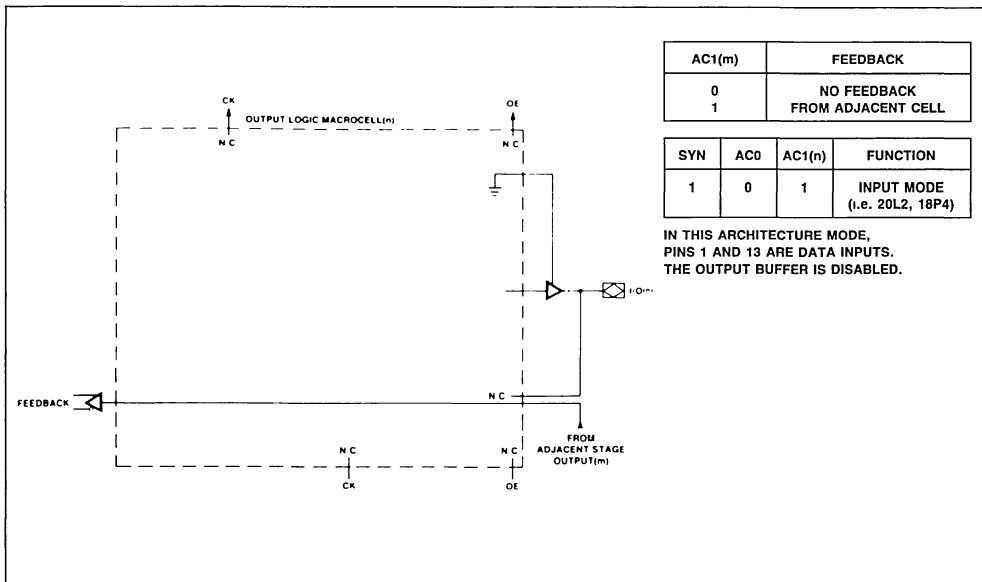


Fig. 7 - Dedicated Input Mode

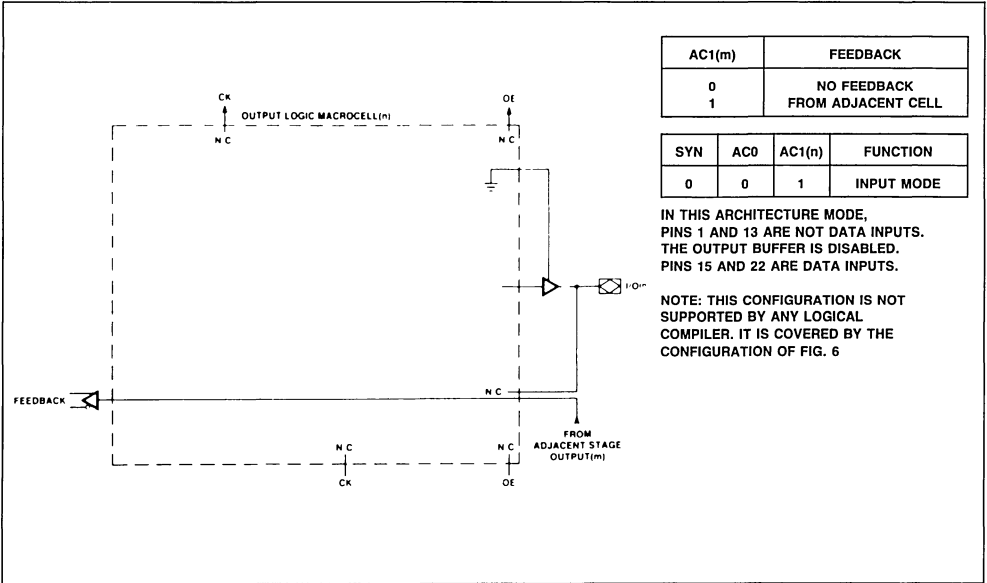
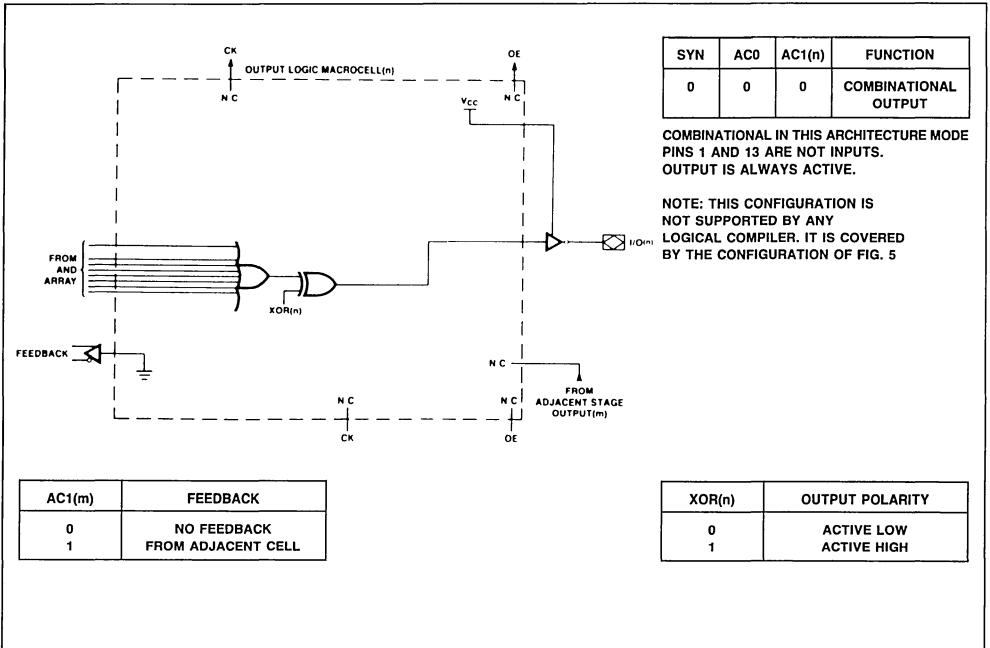


Fig. 8 - Dedicated Combinational Output

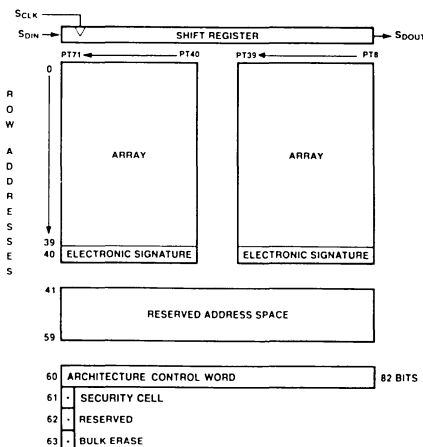


ROW ADDRESS MAP DESCRIPTION

Figure 9 shows a block diagram of the row address map. There are a total of 44 unique row addresses available to the user when programming the GAL20V8 devices. Row addresses 0-39 each contain 64 bits of input term data. This is the user array where the custom logic pattern is programmed. Row 40 is the Electronic Signature Word. It has 64 bits available for any user defined purpose. Row 41-59 are reserved by the manufacturer and are not available to users.

Row 60 contains the architecture and output polarity information. The 82 bits within this word are programmed to configure the device for a specific application. Row 61 contains a one bit security cell that when programmed prevents further programming verification of the array. Row 63 is the row that is addressed to perform a bulk erase of the device, resetting it back to a virgin state. Each of these functions is described in the following sections.

Figure 9. GAL20V8 Row Address Map Block Diagram



ELECTRONIC SIGNATURE WORD

An Electronic Signature Word is provided with every GAL20V8 device. It resides at row address 40 and contains 64 bits of reprogrammable memory that can contain user-defined data. Some uses include user ID codes, revision numbers, or inventory control. This signature data is always available to the user independent of the state of the security Cell.

ARCHITECTURE CONTROL WORD

All of the various configurations of the GAL20V8 devices are controlled by programming cells within the 82-bit Architecture Control Word that resides at row 60. The location of specific bits within the Architecture Control Word is shown in the control word diagram in Figure 10. The function of the SYN, AC0 and AC1(n) bits have been explained in the OUTPUT LOGIC MACROCELL description. The eight polarity bits determine each output's polarity individually. The numbers below the XOR(n) and AC1(n) bits in shows the output device pin number that the polarity bits control.

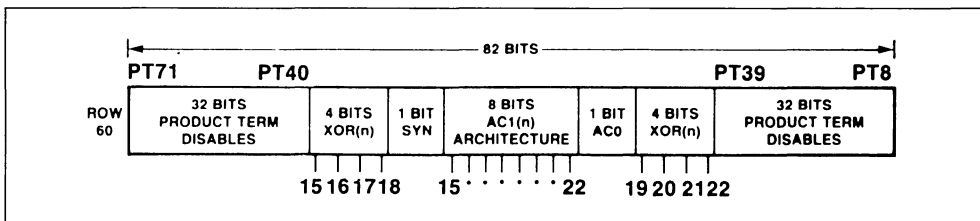
SECURITY CELL

Row address 61 contains the Security Cell (one bit). The Security Cell is provided on all GAL20V8 devices as a deterrent to unauthorized copying of the array configuration patterns. Once programmed, the circuitry enabling array access is disabled, preventing further programming or verification of the array (rows 0-39). The cell can be erased only in conjunction with the array during a bulk erase cycle, so the original configuration can never be examined once this cell is programmed. Signature data is always available to the user.

BULK ERASE MODE

By addressing row 63 during a programming cycle, a clear function performs a bulk erase of the array and the Architecture Control Word. In addition, the Electronic Signature Word and the Security Cell are erased. This mode resets a previously configured device back to its virgin state.

Figure 10. GAL20V8 Architecture Control Word Diagram



OUTPUT REGISTER PRELOAD

When testing state machine designs, all possible states and state transitions must be verified in the design, not just those required in the normal machine operations. This is because in system operation, certain events occur that may throw the logic into an illegal state (power-up, line voltage glitches, brown-outs, etc.). To test a design for proper treatment of these conditions, a way must be provided to break the feedback paths, and force any desired (ie. illegal) state into a registers. Then the machine can be sequenced and the outputs tested for the correct next state conditions.

The GAL20V8 device includes circuitry that allows each registered output to be synchronously set either high or low. Thus, any present state condition can be forced for test sequencing. Figure 11 shows the pin functions necessary to preload the register. The register preload timing and pin voltage levels necessary to perform the function are shown below (see fig. 12). This test mode is entered by raising PRLD to V_{IES} , which enables the serial data in (S_{DIN}) buffer and the serial data out (S_{DOUT}) buffer. Data is then serially shifted into the registers on each rising edge of the clock, DLCK. Only the

macrocells with registered output configurations are loaded. If only 3 outputs have registers, then only 3 bits need be shifted in. The registers are loaded from the bottom up as shown in figure 11.

Figure 11. Output Register Preload Pinout

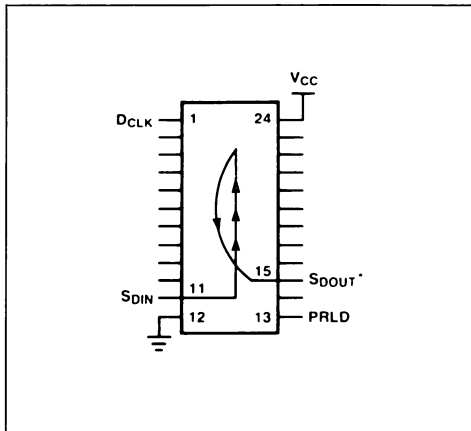


Figure 12. Register Preload Waveforms

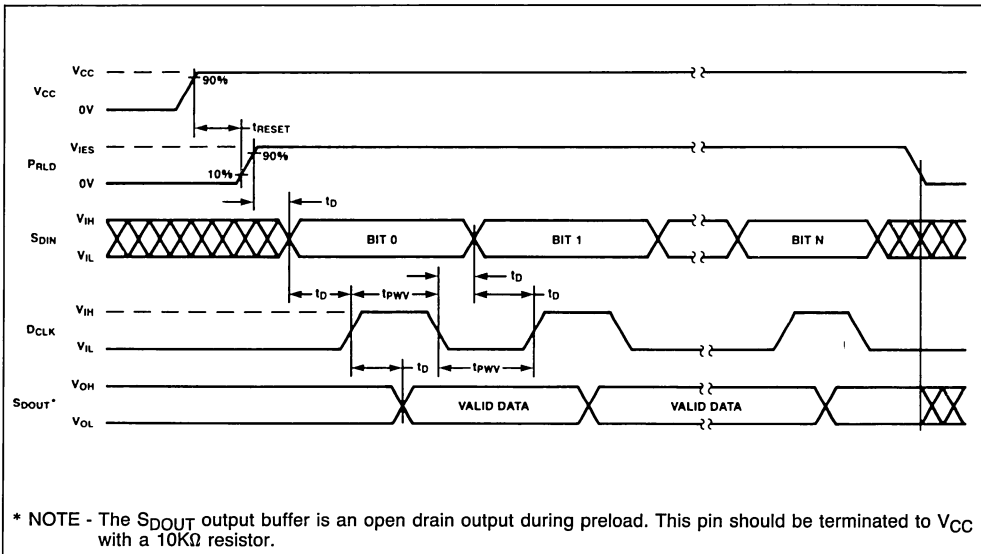
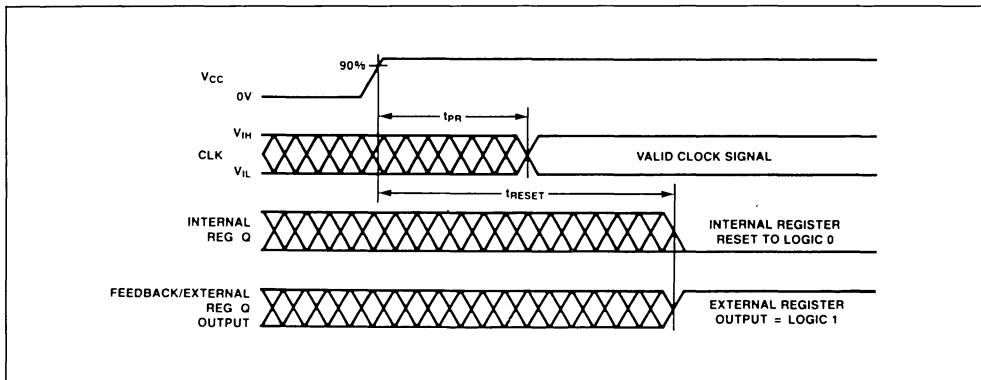


Figure 13. Power-up Reset



LATCH-UP PROTECTION

GAL devices are designed with on board charge pump to negatively bias the substrate. The negative bias is of sufficient magnitude to prevent input undershoots from causing the circuitry to latch. Additionally, outputs are designed with n-channel pullups instead of the traditional p-channel pullups to eliminate any possibility of SCR induced latching.

Circuitry within the GAL20V8 provides a reset signal to all registers during power-up (see fig. 13). All internal registers will have their Q outputs set low after a specified time (t_{RESET}). As a result, the state on the registered output pins (if they are enable through OE) will always be high on power-up, regardless of the programmed polarity of the output pins. This feature can greatly simplify state machine design by providing a known state on power-up.

The timing diagram for power-up is shown above. Because of asynchronous nature of system power-up, some conditions must be met to guarantee a valid power-up reset of the GAL20V8. First, the VCC rise must be monotonic. Second, the clock input must become a proper TTL level within the specified time (t_{PR}). The registers will reset within a maximum of t_{RESET} time. As in normal system operation, avoid clocking the device until all input and feedback path setup times have been met.

FIELD SUPPORT TOOLS

Although it is possible to program GAL devices manually, SGS-THOMSON strongly recommends the use of approved programming hardware and software. programming on unapproved equipment will generally void all guarantees.

PROGRAMMER/DEVELOPMENT SYSTEMS

VENDOR	SYSTEM	REVISION
DATA I/O	Model 29B	V04
	Adapter 303A-011A	V06
	Unisite	V1.7
	Model 60A	V12
Adapter 360-001A		
STAG	Model ZL30A	30A28
Programmable Logic Tech.	Logic Lab	V2.1 (◇)
Qwerty	QPR-1000+	2.0

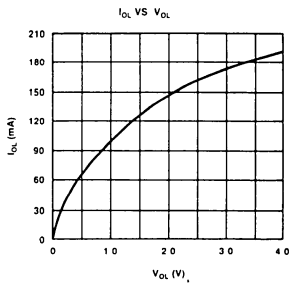
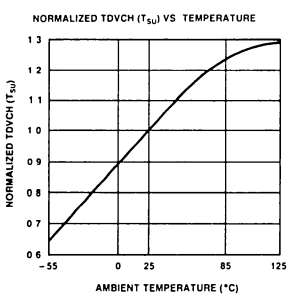
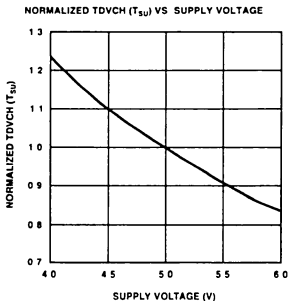
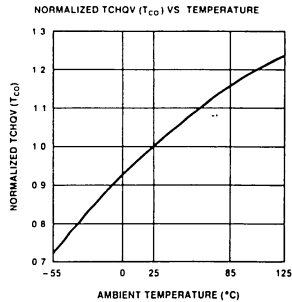
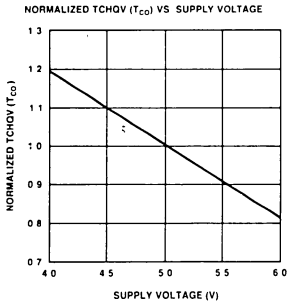
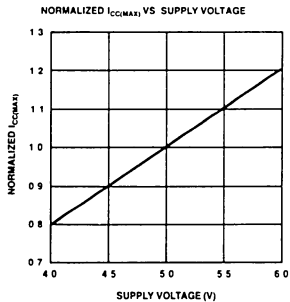
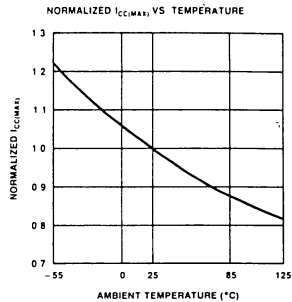
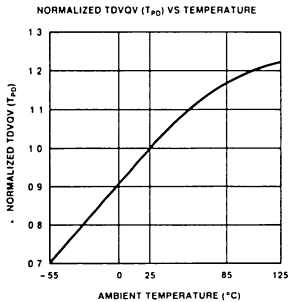
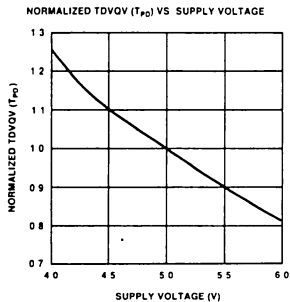
(◇) In conjunction with IBM PC (or compatible) using FastMap Plus* V2.1 software.

SOFTWARE DEVELOPMENT TOOLS†

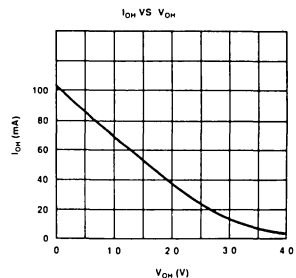
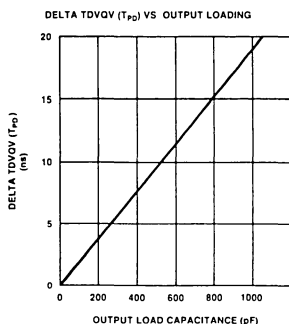
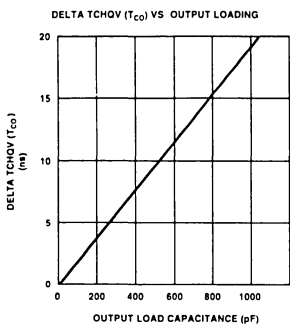
PACKAGE	VENDOR	REVISION
CUPL*	Personal Cad Systems	V2.15
ABEL*	Data I/O	V3.0
PLDtest*	Data I/O	V1.0 (†)
DASH-ABEL	Data I/O	V1.0
PALASM*	Monolithic Memories Inc.	(†)
LC9000*	Programmable Logic Tec.	V1.5
PLAQ*	Qwerty Inc.	V1.0

(†) When emulating PAL devices any revision of the software can be used to create the PAL JEDEC file. Some programming hardware will automatically configure the GAL architecture.

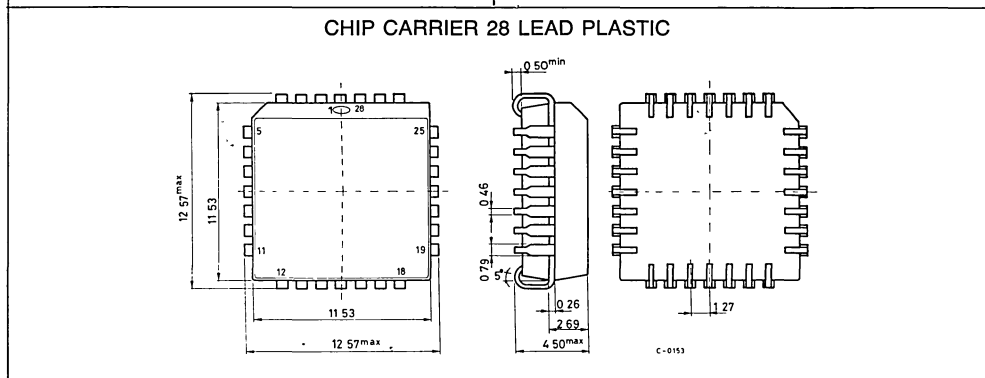
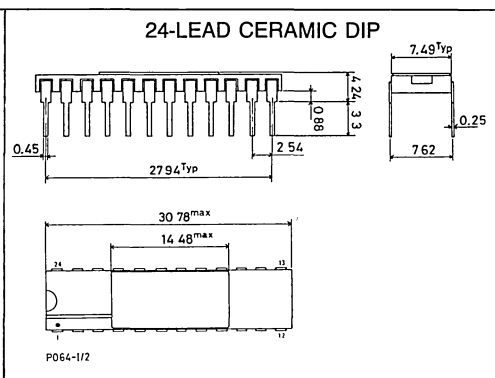
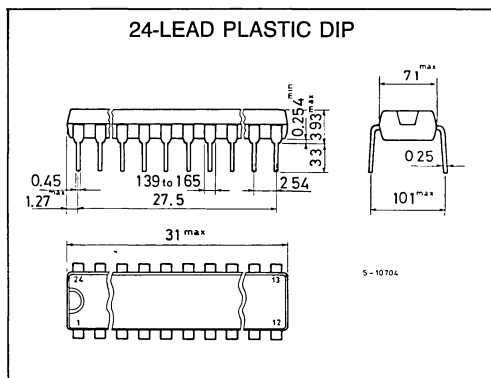
GAL20V8 SPECIFICATION



GAL20V8 SPECIFICATION (Continued)



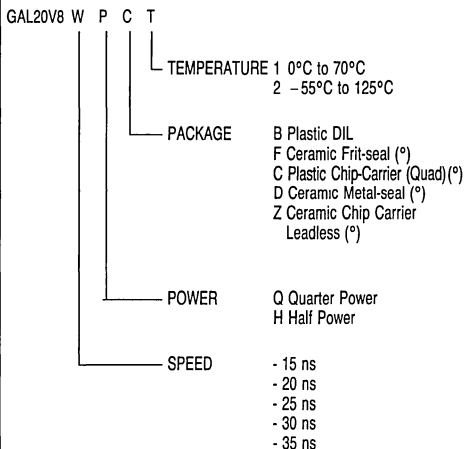
PACKAGE INFORMATION



GAL20V8 SPECIFICATION (Continued)

ORDERING INFORMATION

The SGS-THOMSON GAL devices are available in a variety of packages and two temperature ranges. General ordering code is reported below.



EXAMPLE: Order Code for a GAL20V8, 20 ns speed and half power in plastic DIL is: **GAL20V8-20HB1**

(°) Please contact SGS-THOMSON Sales Office

- Burn-in (168h, 125°C) also available for these products (add suffix "X").
- Military temperature range is available in ceramic packages only.

SPEED/POWER CROSS-REFERENCE GUIDE

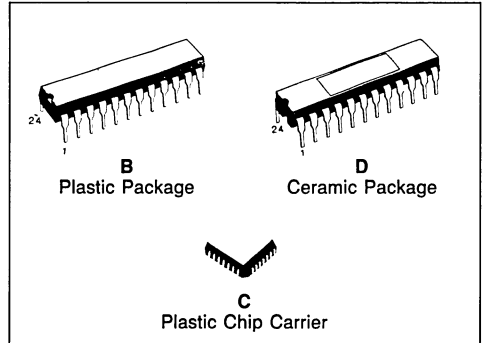
SPEED	POWER	GAL DEVICE	BIPOLAR PAL DEVICE
15ns	45mA	- 15Q	—
15ns	90mA	- 15H	—
15ns	180mA	use - 15H	B
(♦) 20ns	50mA	- 20Q	—
(♦) 20ns	90mA	- 20H	—
(♦) 20ns	210mA	use - 20H	B MIL
25ns	45mA	- 25Q	—
25ns	90mA	- 25H	B-2
25ns	180mA	use - 25H	A
(♦) 30ns	50mA	- 30Q	—
(♦) 30ns	90mA	- 30H	B-2 MIL
(♦) 30ns	210mA	use - 30H	A MIL
35ns	45mA	- 35Q	B-4
35ns	90mA	use - 35Q	A-2
35ns	180mA	use - 35Q	STD
(♦) 40ns	50mA	use - 30Q	B-4 MIL
(♦) 40ns	90mA	use - 30H	A-2 MIL
(♦) 40ns	210mA	use - 30H	STD MIL

(♦) MILITARY TEMPERATURE RANGE

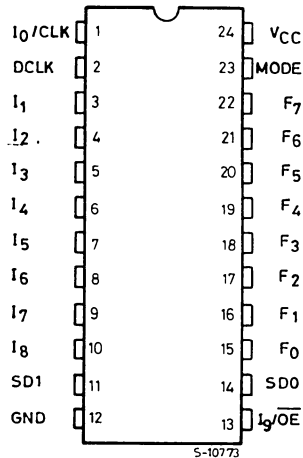
IN-SYSTEM RE-PROGRAMMABLE GENERIC ARRAY LOGIC*

PRODUCT PREVIEW

- **IN-SYSTEM RECONFIGURABLE—5-VOLT ONLY PROGRAMMING**
 - Change Logic "On the Fly" (in less than 1 s)
 - Nonvolatile E² Technology
- **DIAGNOSTICS MODE FOR CONTROLLABILITY AND OBSERVABILITY OF SYSTEM LOGIC**
- **HIGH PERFORMANCE E²CMOS TECHNOLOGY**
 - High Speed: 25ns Max Propagation Delay
 - Low Power: 90mA Max Active
- **EIGHT OUTPUT LOGIC MACROCELLS**
 - Maximum Flexibility for Complex Logic Designs
 - Also Emulates 20-pin PAL Devices with Full Function/Fuse Map/Parametric Compatibility
- **PRELOAD AND POWER-ON RESET OF ALL REGISTERS**
 - 100% Functional Testability
- **SPACE SAVING 24-PIN, 300-MIL DIP**
- **MINIMUM 10,000 ERASE/WRITE CYCLES**
- **DATA RETENTION EXCEEDS 10 YEARS**
- **APPLICATIONS INCLUDE:**
 - Reconfigurable Interfaces
 - Copy Protection and Security Schemes
 - erasable hardware
 - password systems
 - proprietary hardware/software interlocks
 - Configurable memory mapping


DESCRIPTION

The GAL16Z8 (patent pending) is a revolutionary programmable logic device featuring a 5-volt only in system programmability. The device combines a high performance CMOS process with electrically erasable (E²) floating gate technology. This programmable memory technology applied to array logic provides designers with reconfigurable logic and bipolar performance at significantly reduced power levels. The 24-pin GAL16Z8 is architecturally equivalent to the familiar 20 pin GAL 16V8 but includes 4 extra pins to control in-systems programming. It features 8 programmable logic OUTPUT LOGIC MACROCELLs (OLMC) allowing each output to be configured by the user. Additionally, the GAL16Z8 is capable of emulating, in a functional/fuse map compatible device, all 20 pin PAL device architectures (excluding XOR devices). Unique test circuitry and reprogrammable fuses allow complete AC,DC, fuse and functionality testing during manufacture. Therefore, SGS-THOMSON guarantees 100% field programmability and functionality of the GAL devices. In addition, electronic signature is available to provide positive device ID. A security circuit is built-in, providing proprietary designs with copy protection.

PIN CONNECTIONS (top view)


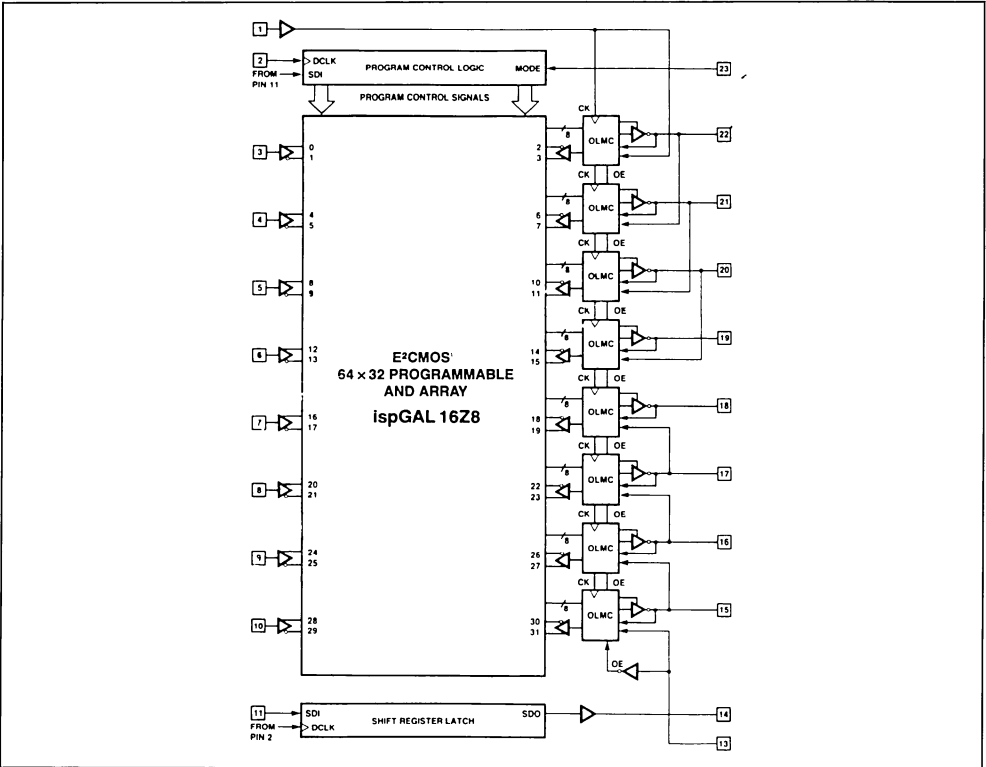
PIN NAMES

I0-I15	INPUT
CK	CLOCK INPUT
DCLK	DATA CLOCK
B0-B5	BIDIRECTIONAL
F0-F7	OUTPUT
$\overline{G(OE)}$	OUTPUT ENABLE
MODE	MODE CONTROL
SDI	SERIAL DATA IN
SDO	SERIAL DATA OUT
V _{CC}	POWER (+5V)
GND	GROUND

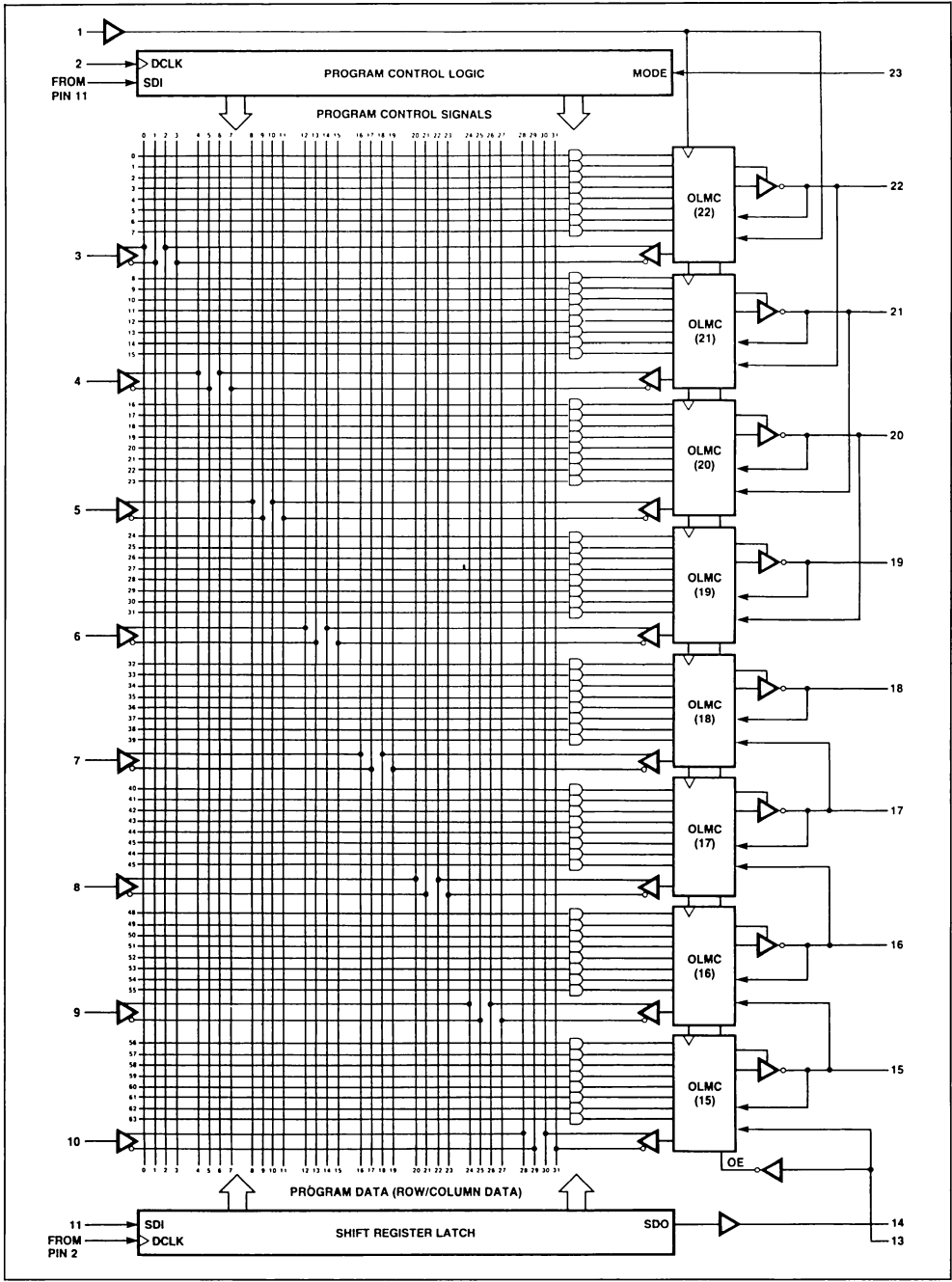
GAL16Z8 REPLACING PAL DEVICES

I ₀ /CK	V _{CC}	V _{CC}	MODE	MODE	MODE	MODE	MODE	MODE	MODE
DCLK	_MODE	MODE	MODE	MODE	MODE	MODE	MODE	MODE	MODE
11	_F7	I11	I13	I15	F7	B1	B1	F1	
12	_F6	F5	I12	I14	F6	F5	B2	B5	
13	_F5	F4	F3	I13	F5	F4	F3	B4	
14	_F4	F3	F2	F1	F4	F3	F2	B3	
15	_F3	F2	F1	F0	F3	F2	F1	B2	
16	_F2	F1	F0	I12	F2	F1	F0	B1	
17	_F1	F0	I11	I11	F1	F0	B1	B0	
18	_F0	I10	I10	I10	F0	B0	B0	F0	
SDI	_SDO	SDO	SDO	SDO	SDO	SDO	SDO	SDO	
GND	_I9	I9	I9	I9	$\overline{G(OE)}$	$\overline{G(OE)}$	$\overline{G(OE)}$	I9	
		10L8	12L6	14L4	16L2	16R8	16R6	16R4	16L8
		10H8	12H6	14H4	16H2	16RP8	16RP6	16RP4	16H8
		10P8	12P6	14P4	16P2				16P8
GAL16Z8									

BLOCK DIAGRAM



LOGIC DIAGRAM



OVERVIEW

The ispGAL16Z8 device has three basic modes of operation: NORMAL, DIAGNOSTIC and PROGRAM. These three modes are controlled by the system designer through the use of a sophisticated on-chip state machine.

In addition, the ispGAL16Z8 has been optimized so that the use of 2 or more devices on a board requires the same amount of control overhead as a single device would. This Serial Loop approach applies to the DIAGNOSTIC and PROGRAM operation modes of the device.

Detailed information on the characteristics, control options and timing for each of the operating modes is available. Contact your sales representative for a copy of this document.

The balance of this document will perform a general review of the operation of the ispGAL16Z8 in its various modes and will explain the use of these control pins.

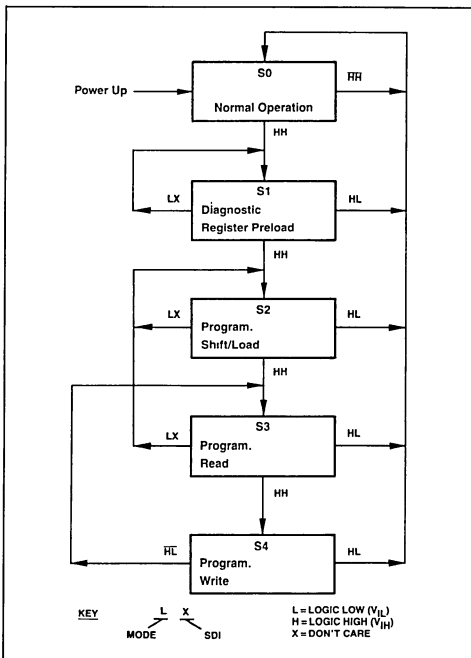
MODE CONTROL & OPERATION

The signals used to control this device are the same for both the DIAGNOSTIC and PROGRAM modes. During NORMAL mode the control pins serve no function other than to control the transition to another mode. The shared control pin approach allows for a simple multi-mode operation with minimal system or board overhead.

The four control signals are TTL level signals; MODE, DCLK, SDI and SDO. These signals are used to transition from mode to mode an through each of the five states as shown in Figure 1. MODE is used only to control the on-chip state machine. DCLK is used for mode control and for the orderly clocking of data into the 16Z8 from the SDI (serial data in) pin as well as out of the device through the SDO (serial data out) pin. SDI is also used for state machine control.

The current state cannot be explicitly observed, however, and "escape" sequence ("HL") to the NORMAL mode is always available to start the process fresh. Caution should be used when using this escape path as the pattern in the device may not be valid if an erase or reprogramming operation was in progress.

Figure 1. On-Chip State Machine



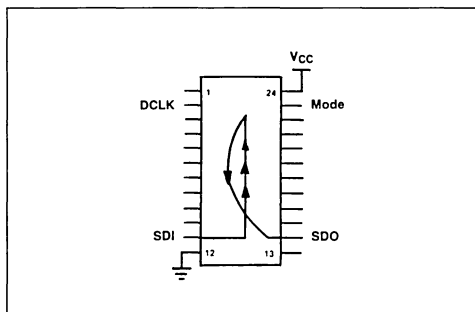
DIAGNOSTIC MODE

From the NORMAL mode the device transitions to the DIAGNOSTIC: Preload state (mode: state). In this mode the values in the Macrocell registers can be interrogated or "pre-"loaded for diagnostic testing of the system. Advanced system design requires full control an observability of all registers on a board.

Upon entry to the DIAGNOSTICS: Preload state the data on the device output pins (15-22) is latched and held to its 1, 0 or three-state condition. This is important as the DIAGNOSTIC: Preload state configures a serial loop from the SDI pin through each of the registers to the SDO pin. Data is shifted across all of the registers during diagnostics (Figure 2). The latching of the current output data insures that the system is not influenced by the changin register contents until such time as the DIAGNOSTIC: Preload state is exited.

The access to the macrocell data is through the SDI and SDO pins. While in the DIAGNOSTIC: Preload state the value in each register is serially shifted out through SDO with each pulse of the DCLK pin. Similarly, new data can be pre-loaded into the registers through the SDI pin.

Figure 2. Diagnostic/Program Pinout



The number of registers in a 16Z8 is a function of the configuration of each macrocell. The length of the serial path, and therefore the number of bits of data shifted in or out of the device, is a function of the number of macrocells which are configured to have registers.

PROGRAM MODE

The PROGRAM mode can only be entered from the DIAGNOSTIC: Preload state. When this transition is made the value of the user programmable Three-State Bit (TSB) is examined to determine the data condition that is held on the device output pins 15-22. The data can either remain latched to 1-0-Z as in the DIAGNOSTIC mode or the data can be forced to high impedance. Again, this feature allows complete control of the system during programming.

The DIAGNOSTIC mode consists of three states of the on-chip state machine: SHIFT, READ and WRITE. Proper sequencing of these states is necessary to program and verify the device. Programming and verification is accomplished using a Serial Register Latch (SRL) to program or verify a row of data at a time.

PROGRA: Shift

During the PROGRAM: Shift state the 88 bit SRL is serially loaded with 82 bits of data and 6 bits of row address for each row to be programmed. The architecture and Electronic Signature of the 16Z8 are also programmed in the same manner. DCLK is used the shift data into SDI for the loading process.

PROGRAM: Read

Verification of data in the array is accomplished in the PROGRAM: Read state. Exiting the PROGRAM: Read state to the PROGRAM: Shift state causes the contents of the array row to be copied to the SRL. This data can be shifted out as outlined above. Programming the Security Cell prevents valid data from being loaded into the SRL. This feature is provided to prevent subsequent copying of the cell patterns.

PROGRAM: Write

The actual programming cycle occurs in the PROGRAM: Write state. The data to be programmed is loaded into the SRL in the PROGRAM: Shift state prior to executing the write cycle. It is the responsibility of the system control logic to assure that the device stays in the PROGRAM: Write state for a sufficient time to program the E² cells, approximately 10 ms. The PROGRAM: Write state is then exited to the PROGRAM: Read state for verification of the data.

The 16Z8 is completely erased by addressing an "erase" row address using the same process outlined above. It is necessary to bulk-erase the device prior to rewriting any pattern into the device as each row write cycle does not include an automatic erase of that row.

The entire programming process takes less than 1/2 second. The bulk erase is 10 ms, and each of the 36 programmable rows can be loaded, programmed and verified in approximately 10.5 ms for a total time of 0.39 seconds. During this time the device output pins are latched or high impedance and the 16Z8 is not responding to changes on its input pins. The system must accommodate this programming time.

SERIAL DIAGNOSTIC/PROGRAM LOOP OPERATION

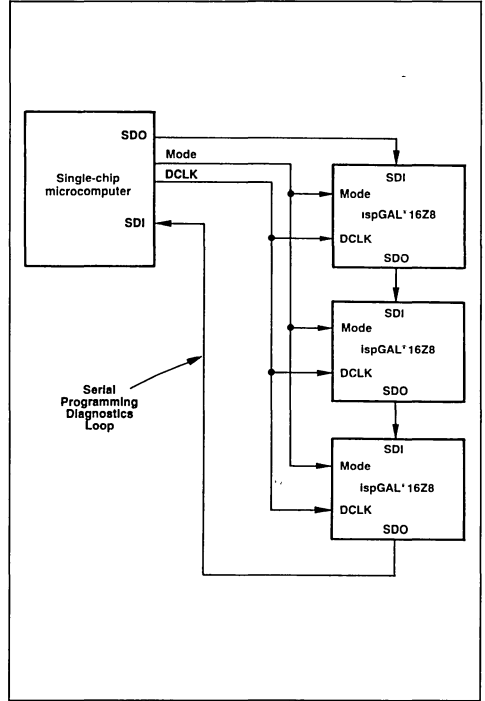
Figure 3 shows a typical ispGAL16Z8 system. Notice that several devices have been cascaded together to form a serial programming loop. This arrangement allows the simultaneous transfer of programming and diagnostic data through every ispGAL device in the system with no additional control logic necessary. When controlling multiple devices in such a loop, the basic diagnostic and programming algorithms remain unchanged. However, there are some additional considerations. In a serial programming loop, the SDO of the first device is connected to the SDI of the second, the SDO of the second to the SDI of the third, and so on. DCLK and MODE are common for every device. With such an arrangement, devices in the loop are always in the same state, but the data being shifted into their respective SRLs may be different. Note that, before data reaches the SDI input of any given device, it must first pass through the 88-bit SRL of every device ahead of it in the loop. The SRL is asynchronously bypassed (SDO = SDI) whenever MODE = 1 allowing SDI to function as both a data and mode control pin.

In a serial diagnostic loop the length of the loops is a function of the number of OLMC registers being used. On the other hand, in a serial programming loop, data transfers always occur in multiples of 88-bits, as the data must pass through the SRL of other devices in the loop.

Because all devices in a loop are always in the same state, reprogramming just one out of "n" devices would seem to be a problem. This, however, is not the case, as several options exist. The most obvious solution is to simply reprogram all of the devices, even if the new pattern is the

same as the old. The system "down" time is effectively the same since all the devices reprogram in parallel.

Figure 3. Typical ispGAL16Z8 System



PROGRAMMER/DEVELOPMENT SYSTEMS

VENDOR	SYSTEM	REVISION
DATA I/O	Model 29B	V04
	Adapter 303A-011A	V06
Programmable Logic Tech.	Logic Lab	V2.20

SOFTWARE DEVELOPMENT TOOLS†

PACKAGE	VENDOR	REVISION
CUPL*	Personal Cad Systems	V2.15
ABEL*	Data I/O	V3.0
PLDtest*	Data I/O	V1.0 (†)
DASH-ABEL	Data I/O	V1.0

(†) When emulating PAL devices any revision of the software can be used to create the PAL JEDEC file. Some programming hardware will automatically configure the GAL architecture.

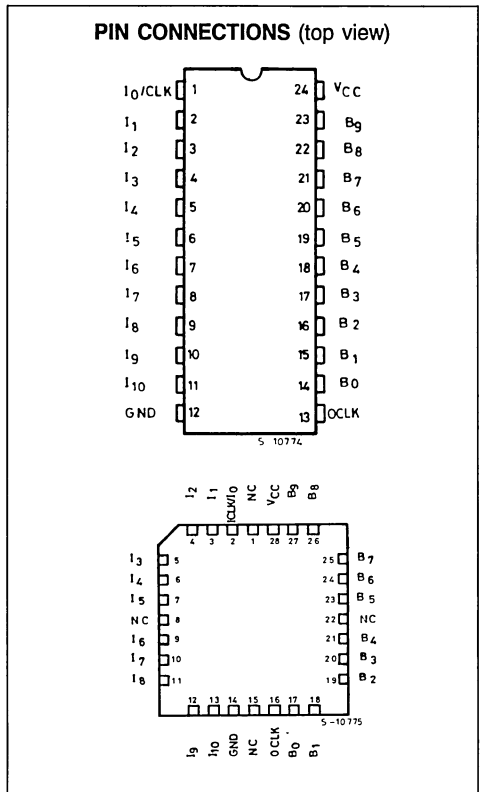
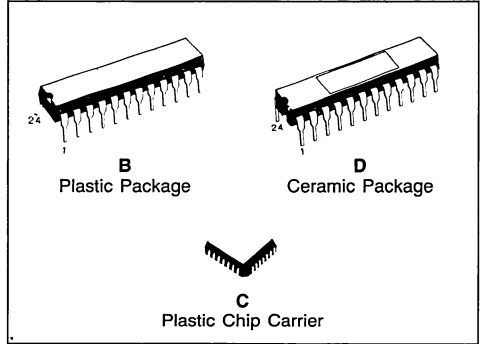
CMOS FPLA GENERIC ARRAY LOGIC

PRODUCT PREVIEW

- **ELECTRICALLY ERASABLE CELL TECHNOLOGY**
 - Instantly Reconfigurable Logic
 - Instantly Reprogrammable Cells
 - Guaranteed 100% Yields.
- **HIGH PERFORMANCE E²CMOS TECHNOLOGY**
 - Low Power: 90mA Typical
 - High Speed: 15ns Max. Clock to Output Delay
25ns Max. Setup Time
30ns Max. Propagation Delay
- **TTL COMPATIBLE INPUTS AND OUTPUTS**
- **UNPRECEDENTED FUNCTIONAL DENSITY**
 - 10 Output Logic Macro Cells
 - 8 State Logic Macro Cells
 - 20 Input and I/O Logic Macro Cells
- **HIGH-LEVEL DESIGN FLEXIBILITY**
 - 78 × 64 × 36 FPLA Architecture
 - Separate State Register and Input Clock Pins
 - Functionally Supersets Existing 24-pin PAL* and IFL* Devices
 - Asynchronous Clocking
- **SPACE SAVING 24-PIN, 300-MIL DIP**
- **HIGH SPEED PROGRAMMING ALGORITHM**
- **20-YEAR DATA RETENTION**

DESCRIPTION

Using a high performance E²CMOS technology, GAL has produced a next generation programmable logic device, the GAL39V18. Using an FPLA architecture known for its superior flexibility in state machine design, the SGS-THOMSON GAL39V18 offers the highest degree of functional integration and flexibility currently available in a 24-pin, 300 mil package. The GAL39V18 has 10 programmable OUTPUT LOGIC MACROCELL (OLMCs) and 8 Programmable "Buried" State Logic Macrocells (SLMCs). In addition, there are 20 Input Logic Macrocells (ILMCs). Two Clock inputs are provided for independent control of the Input and State Macrocells. Advanced features that simplify programming and reduce test time, coupled with E²CMOS reprogrammable cells, enable complete AC, DC, programmability, and functionality test of each GAL39V18 during manufacture. This allows SGS-THOMSON to guarantee 100% field programmability and functionality to data-sheet specifications. Programming is accomplished using standard hardware and software tools. SGS-THOMSON guarantees a minimum of 100 erase write cycles, and data retention to exceed 20 years. An Electronic Signature word has been provided for user-defined data. In addition, a security cell is available to protect proprietary designs.



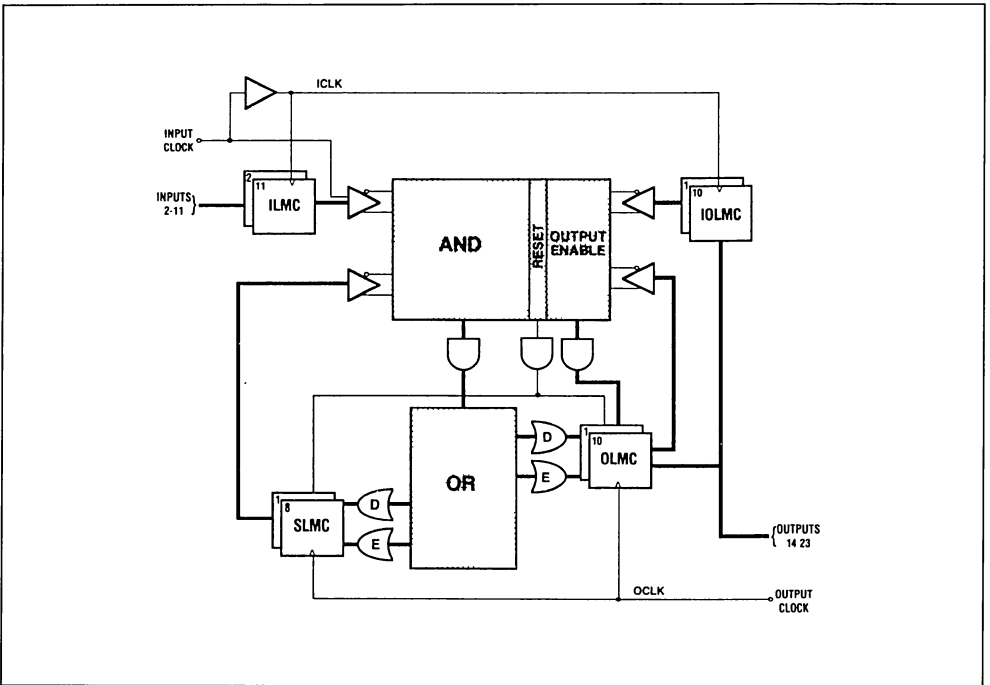
PIN NAMES

I0-I10	INPUT
I _{CLK}	INPUT CLOCK
O _{CLK}	OUTPUT CLOCK
B0-B9	BIDIRECTIONAL
V _{CC}	POWER (+5V)
GND	GROUND

MACRO CELL NAMES

ILMC	INPUT LOGIC MACRO CELL
IOLMC	I/O LOGIC MACRO CELL
SLMC	STATE LOGIC MACRO CELL
OLMC	OUTPUT LOGIC MACRO CELL

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	-0.5 to +7	V
V_I	Input Voltage Applied	-2.5 to $V_{CC} + 1.0$	V
V_O	Off-State Output Voltage Applied	-2.5 to $V_{CC} + 1.0$	V
T_{stg}	Storage Temperature	-65 to 125	°C

Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specification).

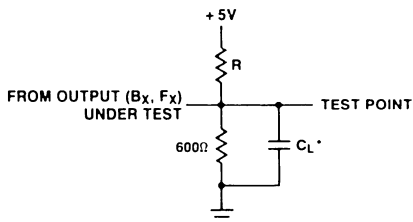
OPERATING RANGE

Symbol	Parameter	Temperature Range						Unit
		Military			Commercial			
		Min.	Typ.	Max.	Min.	Typ.	Max.	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
T_A	Ambient Temperature				0		70	°C
T_C	Case Temperature	-55		125				°C

SWITCHING TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10%-90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See figure

3-state levels are measured 0.5V from steady-state active level.



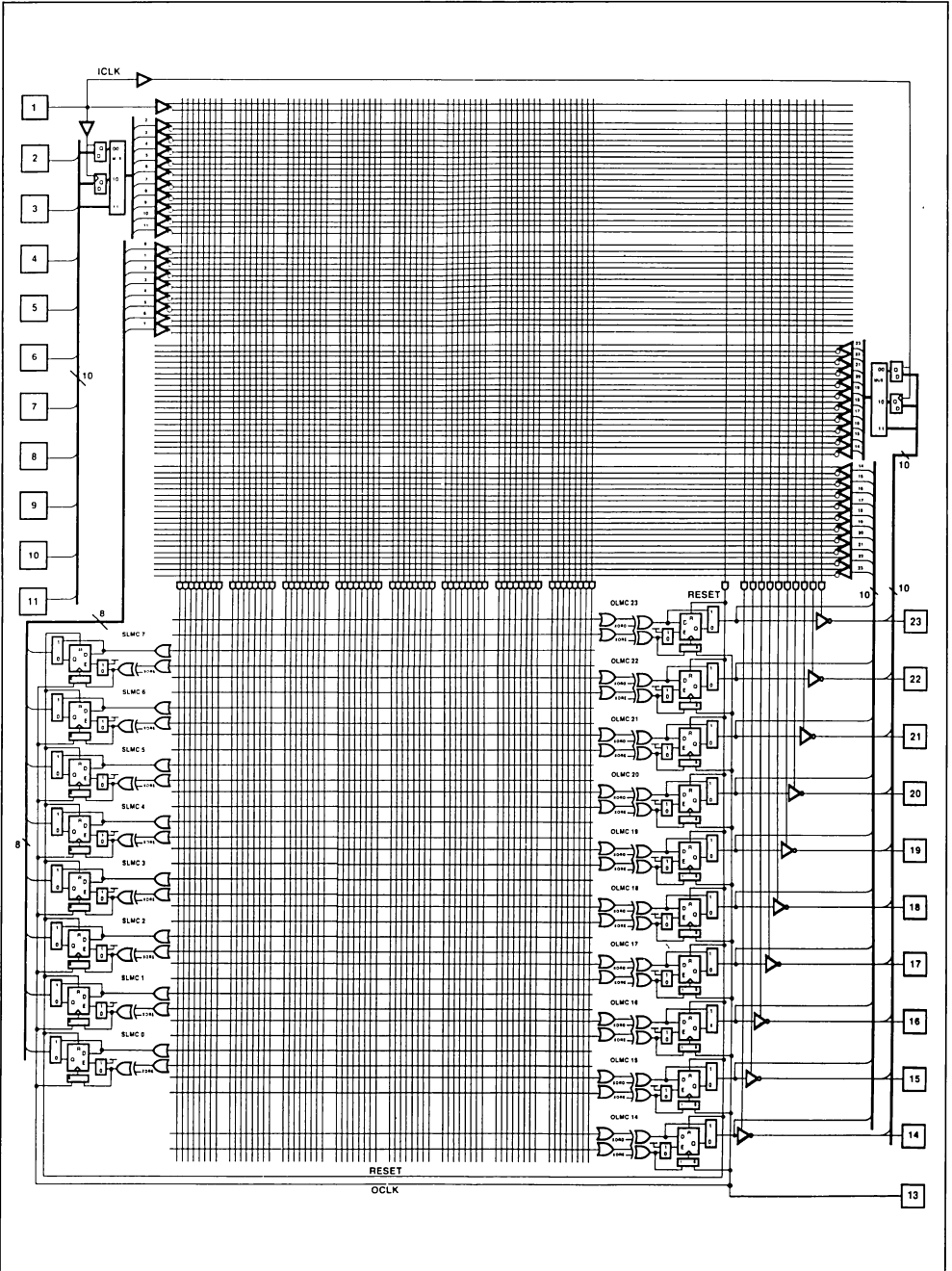
C_L INCLUDES JIG PROBE TOTAL CAPACITANCE

CAPACITANCE ($T_A = 25^\circ\text{C}$, $f = 1.0\text{MHz}$, $V_{CC} = 5\text{V}$)

Symbol	Parameter	Test Conditions	Maximum*	Units
C_I	Input Capacitance	$V_{CC} = 5.0\text{V}$, $V_I = 2.0\text{V}$	12	pF
C_F	Output Capacitance	$V_{CC} = 5.0\text{V}$, $V_F = 2.0\text{V}$	15	pF
C_B	Bidirectional Pin Cap	$V_{CC} = 5.0\text{V}$, $V_B = 2.0\text{V}$	15	pF

* Guaranteed but not 100% tested

LOGIC DIAGRAM



INPUT LOGIC MACRO CELL (ILMC) AND I/O LOGIC MACRO CELL (IOLMC)

The GAL39V18 features two configurable input sections. The ILMC section corresponds to the dedicated input pins (2-11) and the IOLMC to the I/O pins (14-23). Each input section is configurable as a block for asynchronous, latched, or registered inputs. Pin 1 (ICLK) is used as an enable input for latched macrocells (transparent when high) and as a clock for registered macrocells (positive edge triggered).

Configurable input blocks can be used to advantage by system designers. Registered inputs are popular for synchronization and data merging. Transparent latches are useful when the input data is invalid outside a known time window. Direct inputs are used in systems where the input data is well ordered in time. With the GAL39V18, external registers and latches are not necessary.

The various configurations of the input and I/O macrocells are controlled by programming four architecture control bits (INLATCH, INSYN, IOLATCH, IOSYN) within the 68-bit architecture control word. The SYN bits determine whether the macrocells will have register/latch capability or will be strictly asynchronous. The LATCH bits select between latched and registered inputs.

The three valid macrocell configurations are shown in the macrocell equivalent diagrams on the following pages. The truth table associated with each diagram shows the values of the LATCH and SYN bits required to set the macrocell to the configuration shown.

OUPUT LOGIC MACRO CELL (OLMC)/STATE LOGIC MACRO CELL (SLMC)

The outputs of the OR array feed two groups of macrocells. One group of eight macrocells is buried; its output feed back directly into the AND array rather than to device pins. These cells are called the State Logic Macrocells (SLMC), as they are useful for building state machines. The second group of macrocells consists of 10 cells whose outputs, in addition to feeding back into the AND array, are available at the device pins. Cells in this group are known as Ouput Logic Macrocells (OLMC). Like the ILMC and IOLMC discussed above, output and state logic macrocells are configured by programming specific bits in the architecture control word (CKS(i), OUTSYN(i), OXRD(i), XORE(i)), but unlike the input macrocells which must be configured in blocks, these macrocells are configurable on a macrocell-by-macrocell basis. Throughout this data sheet, $i = [14...23]$ for OLMCs and $i = [0...7]$ for SLMCs.

State and Ouput Logic MacroCells may be set to

one of three valid configurations: combinational, D-type registered with sum term (asynchronous) clock, or D/E-type registered. Output macrocells always have I/O capability, with directional control provided by the 10 output enable (OE) product terms. Additionally, the polarity of each OLMC output is selectable through the XORD(i) architecture bits. Polarity selection is not necessary for SLMCs, since both the true and complemented forms of their outputs are available in the AND array. Polarity of all "E" sum terms is selectable through the XORE(i) architecture control bits.

When $CKS(i) = 1$ and $OUTSYN(i) = 0$, macrocell "i" is set as "D/E-type registered." In this configuration, the register is clocked from the common OCLK and the register clock enable input is controlled by the associated "E" sum term. This configuration is useful for building counters and state-machines with state hold functions.

When $CKS(i) = 0$ and $OUTSYN(i) = 0$, macrocell "i" is set as "D-type registered with sum term clock." In this configuration, the register is enabled and its "E" sum term is routed directly to the clock input. This allows for the popular "asynchronous programmable clock" feature, selectable on a register-by-register basis.

When $CKS(i) = 0$ and $OUTSYN(i) = 1$, macrocell "i" is set as "combinational." Configuring a SLMC in this manner turns it into a complement array. Complement arrays are used to construct multi-level logic.

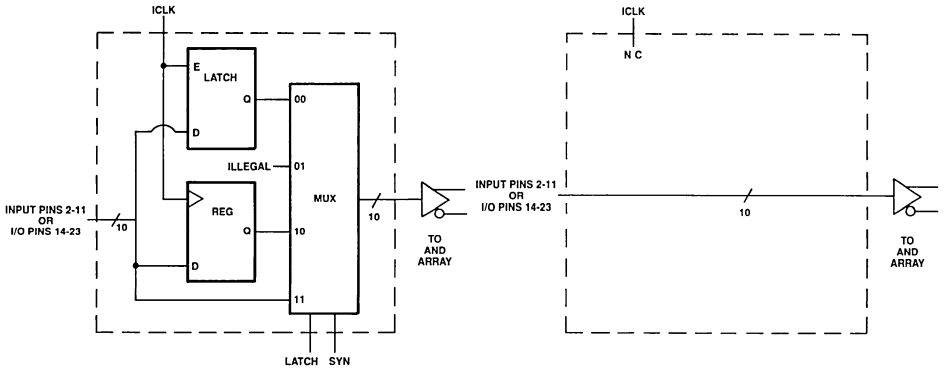
Registers in both the Ouput and State Logic Macrocells feature a RESET input. This active high input allows the registers to be simultaneously and asynchronously reset from a common signal. The source of this signal is the RESET product term. Registers reset to a logic zero, but since the output buffers invert, a logic one will be present at the device pins.

There are two possible feedback paths from each OLMC: one from before the output buffer (this is the normal path), and one from after the output buffer, through the IOLMCs. The second path is usable as a feedback only when the associated bidirectional pin is being used as an output; during input operations it becomes the input data path, turning the associated OLMC into an additional buried state macrocell.

The D/E registers used in this device offer the designed the ultimate in flexibility and utility. The D/E register construct can emulate RS-, JK-, and T-type registers with the same efficiency as a dedicated RS-, JK-, or T-register.

The three valid macrocell configurations are shown in the macrocell equivalent diagrams on the following pages. The truth table associated with each diagram shows the bit value of CKS(i) and OUTSYN(i) required to set the macrocell to the configuration shown.

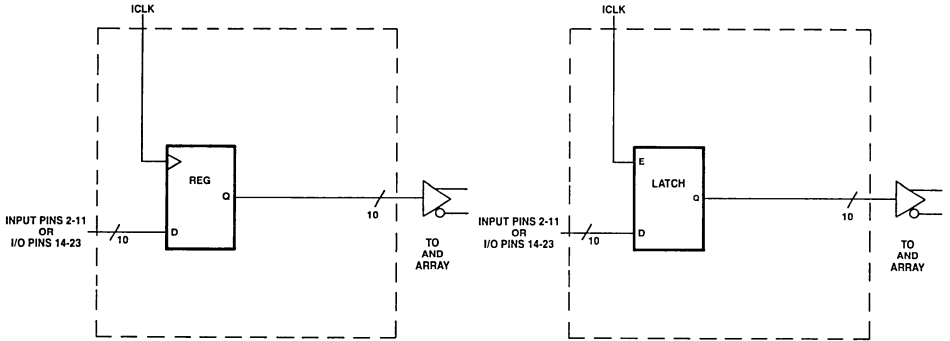
ILMC/IOLMC CONFIGURATIONS



ILMC/IOLMC
Generic Block Diagram

Asynchronous Input

LATCH	SYN
1	1



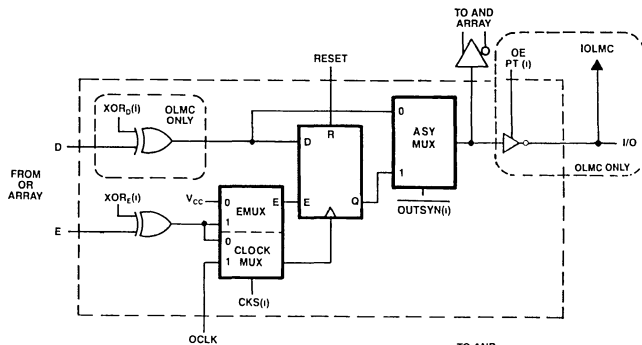
Registered Input

LATCH	SYN
1	0

Latched Input

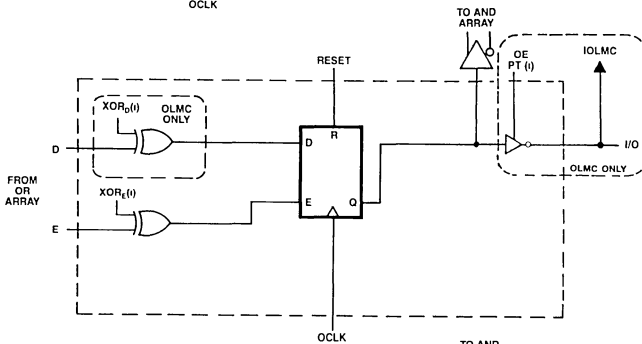
LATCH	SYN
0	0

OLMC/SLMC
Block Diagram



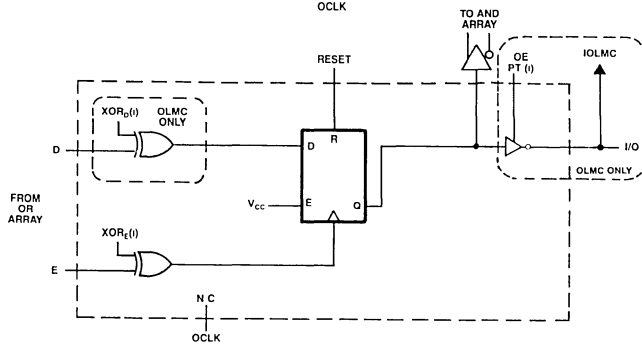
D/E Type Registered

CKS(i)	OUTSYN(i)
1	0



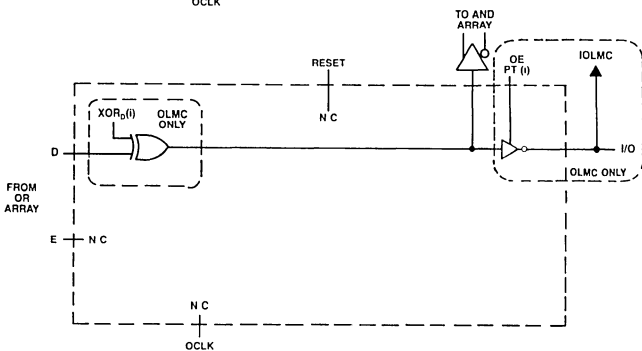
D-Type Registered with Sum Term Asynchronous Clock

CKS(i)	OUTSYN(i)
0	0



Combinational

CKS(i)	OUTSYN(i)
0	1



ARRAY DESCRIPTION

The GAL39V18 E² reprogrammable array is subdivided into three smaller arrays: AND, OR, and Architecture. These arrays are described in detail below.

AND ARRAY

The AND array is organized as 78 input terms by 75 product term outputs. The 20 input and I/O logic macrocells, 8 SLMC feedbacks, 10 OLMC feedbacks, and ICLK comprise the 39 inputs to this array (each available in true and complemented forms). Product terms 0-63 serve as inputs to the OR array. Product term 64 is the RESET PT; it generates the RESET signal described in the earlier discussion of output and state logic macrocells. Product terms 65-74 are the output enable product terms; they control the output buffers, thus enabling device pins 14-23 to be bi-directional or 3-state.

OR ARRAY

The OR array is organized as 64 inputs by 36 sum term outputs. Product terms 0-63 of the AND array serve as the inputs to this array. Of the 36 sum term outputs, 18 are data ("D") terms and 18 are enable/clock ("E") terms. These terms feed into the 10 OLMCs and 8 SLMCs, one "D" term and one "E" term to each.

ARCHITECTURE ARRAY

The various configurations of the GAL39V18 are enabled by programming cells within the architecture control word. This 68-bit word contains all of the chip configuration data. This data includes: XORD(i), XORE(i), CKS(i), OUTSYN(i), INLATCH, INSYN, IOLATCH, and IOSYN. The function of each of these bits has been previously explained.

ELECTRONIC SIGNATURE WORD

Every GAL39V18 device contains an electronic signature word. The Electronic Signature word is a 72-bit user definable storage area, which can be used to store inventory control data, pattern revision numbers, manufacture date, etc. Signature data is always available to the user, regardless of the state of the security cell.

SECURITY CELL

A security cell is provided with every GAL39V18 device as a deterrent to unauthorized copying of

the array patterns. Once programmed, this cell prevents further read access to the AND, OR, and architecture arrays. This cell can be erased only during a bulk erase cycle, so the original configuration can never be examined once this cell is programmed. Electronic Signature data is always available to the user, regardless of the state of this control cell.

BULK ERASE

Before writing a new pattern into a previously programmed part, the old pattern must first be erased. This erasure is done automatically by the programming hardware as part of the programming cycle and takes only 50 milliseconds.

REGISTERED PRELOAD

When testing state machine designs, all possible states and state transitions must be verified, not just those required during normal machine operations. This is because in system operation, certain events may occur that cause the logic to assume an illegal state: power-up, brown out, line voltage glitches, etc. To test a design for proper treatment of these conditions, a method must be provided to break the feedback paths and force any desired state (i.e., illegal) into the registers. Then the machine can be sequenced and the outputs tested for correct next state generation.

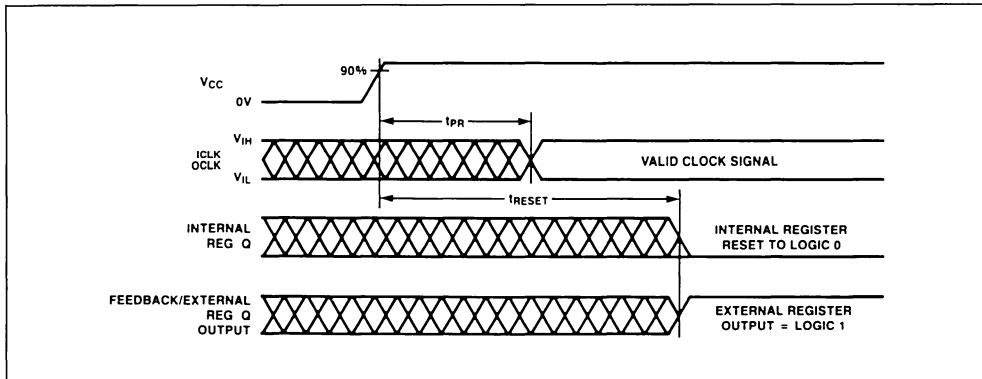
All of the registers in the GAL39V18 can be preloaded, including the input, I/O, and state registers. In addition, the contents of the state and output registers can be examined in a special diagnostics mode. Programming hardware takes care of all preload timing and voltage requirements.

INPUT BUFFERS

GAL devices are designed with TTL level compatible input buffers. These buffers, with their characteristically high impedance, load driving logic much less than "traditional bipolar devices". This allows for a greater fan out from the driving logic.

GAL devices do not possess active pull-ups within their input structures. As a result, Lattice recommends that all unused inputs and 3-state I/O pins be connected to another active input, V_{CC}, or GND. Doing this will tend to improve noise immunity and reduce I_{CC} for the device.

POWER-UP RESET



Circuitry within the GAL39V18 provides a reset signal to all registers during power-up. All internal registers will have their Q outputs set low after a specified time (t_{RESET}). As a result, the state on the registered output pins (if they are enabled) will always be high on power-up, regardless of the programmed polarity of the output pins. This feature can greatly simplify state machine design by providing a known state on power-up.

The timing diagram for power-up is shown above. Because of the asynchronous nature of system power-up, some conditions must be met to guarantee a valid power-up reset of the GAL39V18. First, the V_{CC} rise must be monotonic. Second, the clock inputs must be a proper TTL level within the specified time (t_{PR}). The registers will reset within a maximum of t_{RESET} time. As in normal system operation, avoid clocking the device until all input and feedback path setup times have been met.

PROGRAMMER/DEVELOPMENT SYSTEMS

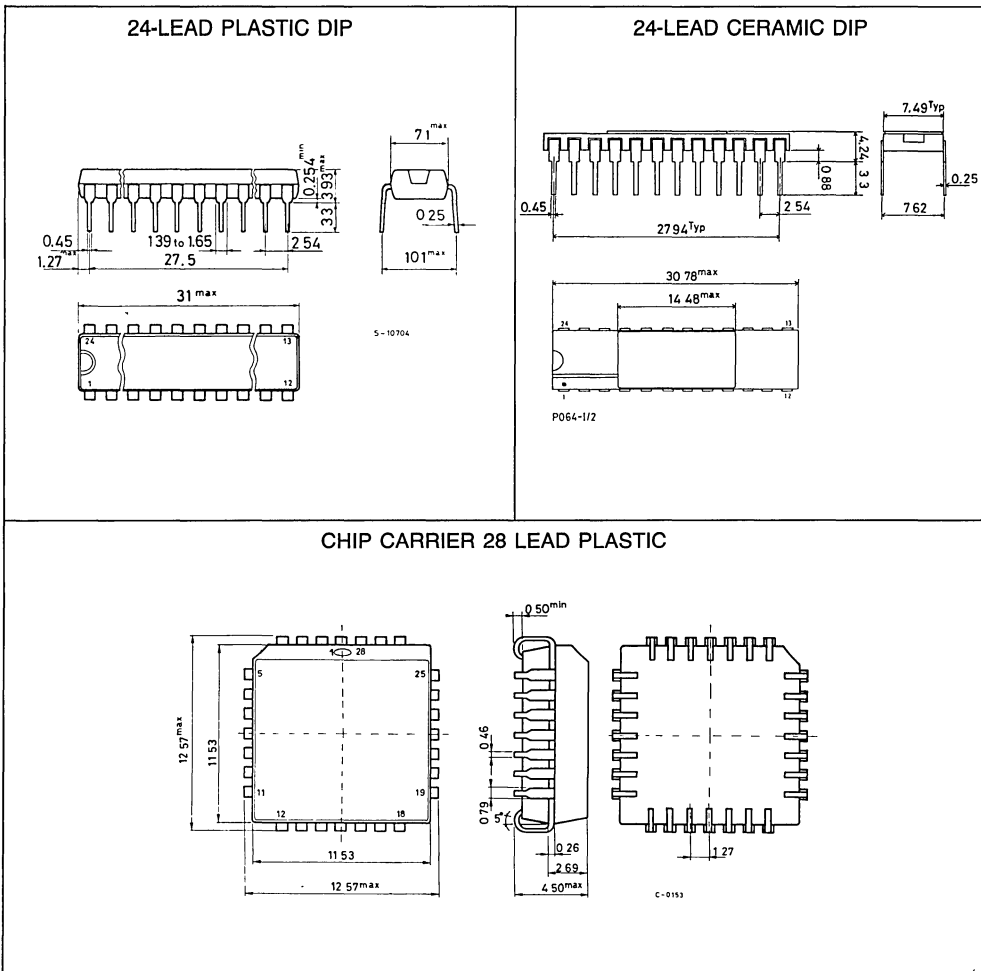
VENDOR	SYSTEM	REVISION
DATA I/O	Model 29B	V04
	Adapter 303A-011A	V06
Programmable Logic Tech.	Logic Lab	V2.20

SOFTWARE DEVELOPMENT TOOLS†

PACKAGE	VENDOR	REVISION
CUPL*	Personal Cad Systems	V2.15
ABEL*	Data I/O	V3.0
LC9000*	Programmable Logic Tec.	V1.5
PLAQ*	Qwerty Inc.	V1.0

(†) When emulating PAL devices any revision of the software can be used to create the PAL JEDEC file. Some programming hardware will automatically configure the GAL architecture.

PACKAGE INFORMATION



GAL39V18 SPECIFICATION (Continued)

ORDERING INFORMATION

The SGS-THOMSON GAL devices are available in a variety of packages and two temperature ranges. General ordering code is reported below.

GAL39V18	W	P	C	T	
					TEMPERATURE 1 0°C to 70°C
					2 -55°C to 125°C
					PACKAGE
					B Plastic DIL
					F Ceramic Frit-seal (°)
					C Plastic Chip-Carrier (Quad)(°)
					D Ceramic Metal-seal (°)
					Z Ceramic Chip Carrier Leadless (°)
					POWER
					Q Quarter Power
					H Half Power
					SPEED
					- 15 ns
					- 20 ns
					- 25 ns
					- 30 ns
					- 35 ns

EXAMPLE: Order Code for a GAL39V18, 20 ns speed and half power in plastic DIL is: **GAL39V18-20HB1**

(°) Please contact SGS-THOMSON Sales Office

- Burn-in (168h, 125°C) also available for these products (add suffix "X").
- Military temperature range is available in ceramic packages only.

SPEED/POWER CROSS-REFERENCE GUIDE

SPEED	POWER	GAL DEVICE	BIPOLAR PAL DEVICE
15ns	45mA	- 15Q	—
15ns	90mA	- 15H	—
15ns	180mA	use - 15H	B
(♦) 20ns	50mA	- 20Q	—
(♦) 20ns	90mA	- 20H	—
(♦) 20ns	210mA	use - 20H	B MIL
25ns	45mA	- 25Q	—
25ns	90mA	- 25H	B-2
25ns	180mA	use - 25H	A
(♦) 30ns	50mA	- 30Q	—
(♦) 30ns	90mA	- 30H	B-2 MIL
(♦) 30ns	210mA	use - 30H	A MIL
35ns	45mA	- 35Q	B-4
35ns	90mA	use - 35Q	A-2
35ns	180mA	use - 35Q	STD
(♦) 40ns	50mA	use - 30Q	B-4 MIL
(♦) 40ns	90mA	use - 30H	A-2 MIL
(♦) 40ns	210mA	use - 30H	STD MIL

(♦) MILITARY TEMPERATURE RANGE

LOGIC CONCEPT

LOGIC DESIGN BASIC CONCEPTS

The digital logic design process is based on Boolean algebra. Here we deal only with the fundamentals of Boolean algebra necessary to implement basic logic functions in a programmable logic device.

Boolean Algebra - Basic Functions

Boolean algebra is based on the "yes" or "no" conditions that can also be expressed as "true" or "false", "high" or "low", or in digital electronic terms as "1" or "0".

All the operations in Boolean algebra are expressed in terms of these two states through the three basic functions: AND, OR and NOT summarized in Figure 1.

Except for the NOT operator, the "AND" and "OR" follow the basic properties of algebra:

- (1) $C = \bar{A} + \bar{B}$
- (2) $D = (A + B)$

In case 1, values of A and B are inverted before evaluating the OR. In the case 2 OR has to be evaluated before the inverting operation because of the parenthesis.

Commutative Property:

$$X * Y = Y * X$$

$$X + Y = Y + X$$

Associative Property:

$$(X + Y) + Z = X + (Y + Z)$$

$$(X * Y) * Z = X * (Y * Z)$$

POSTULATES

Additive and Multiplicative Identity Elements:

$$X + 0 = X$$

$$X * 1 = X$$

Distributivity:

$$X + (Y * Z) = (X + Y) * (X + Z)$$

$$X * (Y + Z) = (X * Y) + (X * Z)$$

Complementation:

$$X + \bar{X} = 1$$

$$X * \bar{X} = 0$$

THEOREMS

Idempotence:

$$X * X = X$$

$$X + X = X$$

Special properties of 0 and 1:

$$X * 0 = 0 \quad \bar{0} = 1$$

$$X + 1 = 1 \quad \bar{1} = 0$$

$$0 * 0 = 0 \quad 1 + 1 = 1$$

$$0 + 0 = 0 \quad 1 * 1 = 1$$

Absorption:

$$X * (X + Y) = X$$

$$X + (X * Y) = X$$

DeMorgan's Law:

$$\overline{(X + Y + Z + \dots)} = \bar{X} * \bar{Y} * \bar{Z} * \dots$$

$$\overline{(X * Y * Z * \dots)} = \bar{X} + \bar{Y} + \bar{Z} + \dots$$

Also very important in Boolean algebra is the Duality principle that states that if the following replacements are made, a logical equivalent expression can be generated:

- Replace all TRUES with FALSES
- Replace all FALSES with TRUES
- Replace all ANDs with ORs
- Replace all ORs with ANDs

Figure 1. Boolean Operators

LOGIC SYMBOL	TRUTH TABLE	EQUATION	FUNCTION	LOGIC OPERATOR															
	<table border="1"> <tr><td>A</td><td>B</td><td>C</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	C	0	0	0	0	1	1	1	0	1	1	1	1	$C = A + B$	OR	"+"
A	B	C																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
	<table border="1"> <tr><td>A</td><td>B</td><td>C</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	C	0	0	0	0	1	0	1	0	0	1	1	1	$C = A * B$ or $C = A \cdot B$ or $C = AB$	AND	"*" or "•"
A	B	C																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
	<table border="1"> <tr><td>A</td><td>C</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	C	0	1	1	0	$C = \bar{A}$ $C = /A$	NOT	"/'" or "'_'"									
A	C																		
0	1																		
1	0																		

Karnaugh Maps

Once we define the function we want to implement, it is very important to optimize it using the theorems already illustrated or using the visual tool known as Karnaugh maps. This map aids in the reduction of logic functions to either of two special formats that are easily transferred into PLD logic maps: these formats are the Sum Of Products (SOP) and the Product Of Sums (POS).

The goal of the designer of the development software is to transform the logic definition into an acceptable format. The POS format can be used to describe any combinatorial logic function. This two-level format consists of logical OR terms that are ANDed together. Thus

$$(1) Y = A * (C + D) + B * C + B * D$$

can be simplified to:

$$(2) Y = A * (C + D) + B * (C + D)$$

and then

$$(3) Y = (A + B) * (C + D)$$

which is an AND of sum terms.

The most common representation is the dual of the Product Of Sums format and is known as the Sum Of Products. The basic PLD array interconnects are of this form. The Sum Of Products (SOP) consists of several AND terms ORed (summed) together. Eq. (1) can also be simplified to a SOP form:

$$(4) Y = A * (C + D) + B * C + B * D$$

and then

$$(5) Y = A * C + A * D + B * C + B * D$$

The above transformations are shown in Figure 2

DeMorgan's Law

A closer examination of the above two implementations of the same function in POS (Eq. (3)) and SOP (Eq. (5)) forms shows that the number of terms feeding into the final gate varies with the implementation. The SOP format required only 4 terms. This observation is critical, since the total number of terms in any PLD is limited. De Morgan's law (defined earlier in this section) is a simple rule that can quickly convert SOP to POS or back, without altering the final logic function. This can allow the number of terms to be reduced by as much as 50%, to overcome device limitations. Eq. (3) can be converted to SOP as follows:

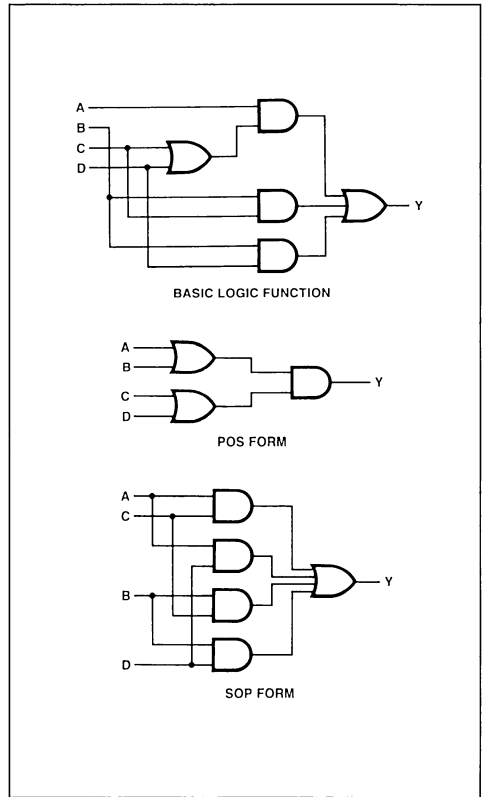
$$(3) Y = (A + B) * (C + D)$$

using duality:

- first step is to change all "Trues" to "Falses"

$$(4) \bar{Y} = (\bar{A} + \bar{B}) * (\bar{C} + \bar{D})$$

Figure 2. Basic Function Formats



- second step is to change all "ANDs" to "ORs" and all "ORs" to "ANDs"

$$(5) \bar{Y} = (\bar{A} * \bar{B}) + (\bar{C} * \bar{D})$$

Eq. (5) is the SOP form. Note, however, that the output function is inverted (or "active low"). A subsequent inversion function will be required to produce the original output function.

Reduction of Equations

Generally a complex logic function must be represented in a specific and reduced format to be implemented into a PLD. The various methods touched on above; Karnaugh maps, DeMorgan's law are used to manipulate the equations in conjunction with the basic Postulates and Theorems.

Current generation software handles all equation minimization and will actually allow the use of higher level functions, such as state description and macro functions, as input functions.

PLD Representation

Let us now have a look at the logic conventions used to describe PLD devices. A typical PLD input buffer is shown in fig. 3. Its two outputs are the true and complement of the input.

Figure 3. PLD Input Buffer

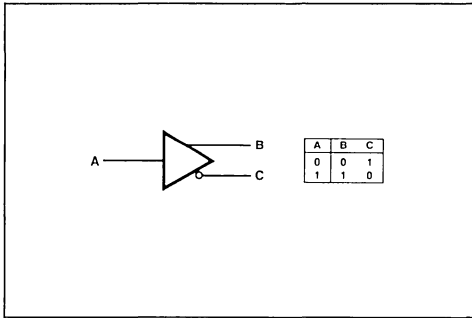
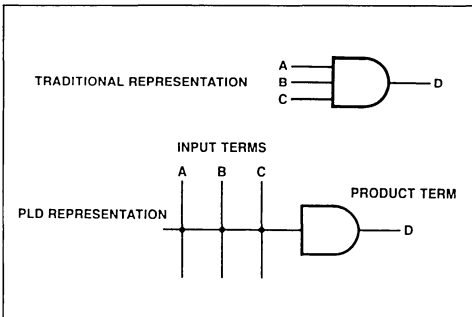


Figure 4 illustrates the convention used to reduce the complexity of a logic diagram without sacrificing any of the clarity. The traditional representation of an AND gate shows three inputs: A, B and C.

Figure 4. AND Gate Representations



The PLD representation has the same three inputs. This shorthand reflects the three distinct input terms of the prior drawing. The structure of a multiple-input AND gate is known as a Product Term.

Referring to Figure 5, we see that the solid-dot connection in the previous figure represents a permanent connection. A programmable interconnection would appear as an X over the intersection, as shown. The X implies that the connection is intact, whereas the absence of an X implies no connection.

Figure 5. PLD Connections

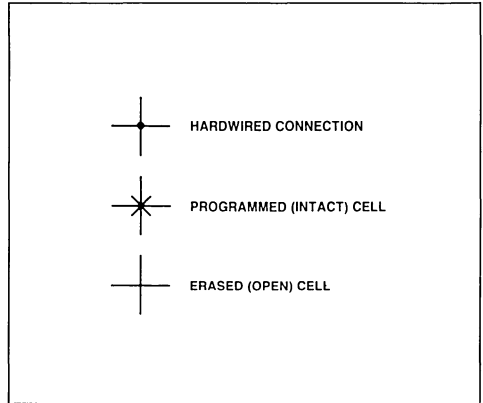
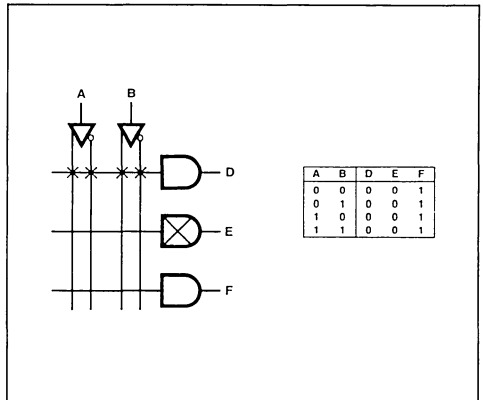


Figure 6 details the default conditions for AND gates.

Figure 6. AND Gate Default Conditions



From the diagram, you can see that the AND gate for output D is connected to all the input terms. The equation for D is

$$D = A * \bar{A} * B * \bar{B}$$

which can be simplified using Boolean algebra

$$D = (A * \bar{A}) * (B * \bar{B})$$

$$D = (0) * (0)$$

$$D = 0$$

The connection of both the true and complement of a given input buffer to a single product term results in that product term always being a logic 0.

PLD Representation (continued)

A shorthand notation for leaving all of the input buffers connected is illustrated on output E. Since logic diagram maps are usually supplied without any of the connections shows as intact, it is much simpler for the designer to connect a whole product term (the device default) by simply drawing "Xs" within the AND gate. Again, this product term will always be a logic 0.

Output F, in contrast, does not have any input terms connected to its product term. This product term will always float to a logic 1, resulting in a 1 on the output. In the following sections, where various PLD architectures will be examined in detail, we will see why this design practice is not recommended.

PAL Devices

The PAL structure in Fig. 7 shows that a PAL consists of a programmable AND array that feeds a fixed OR array. This approach offers the highest performance and the most efficient architecture for most logic functions.

The quantity of product terms per output is fixed by the hardwired OR array. The typical logic function requires some 3 to 4 product terms, well under the 7 to 8 available on current generation devices.

PAL device architectures (the number of inputs, outputs and product terms) have been fixed by the manufacturer, based on a guess as to what the designer may ultimately want. However, the dozens of device types introduced over the last 8 years essentially offer various permutations of three basic output structures.

The first is shown in Figure 8, which illustrates an input and an output with six product terms. The output is always enabled and active low (notice the invert at the OR gate). The true and complement of the input are available to the AND array.

Figure 7. Basic PAL Device Architecture

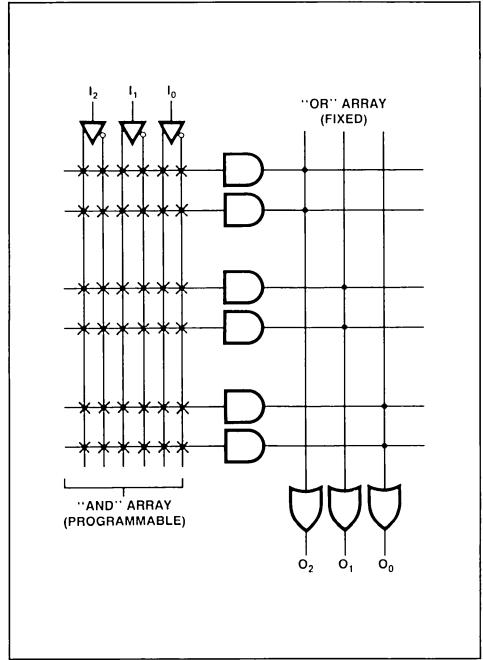
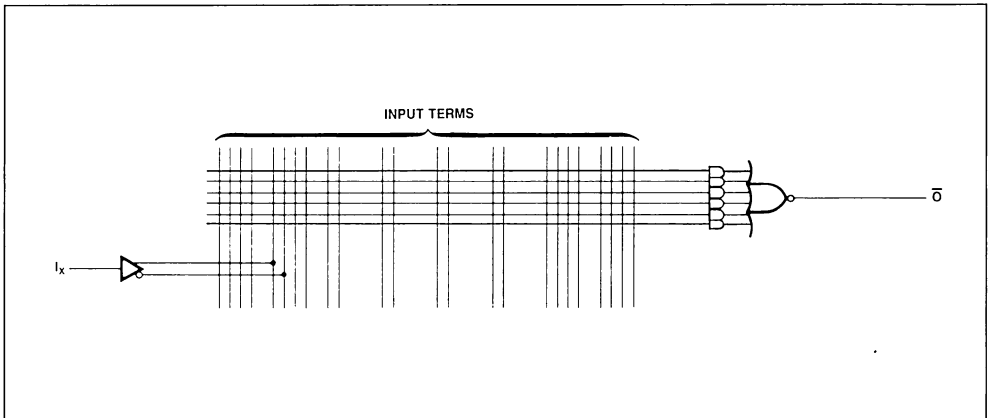


Figure 8. Dedicated Output Structure



PAL Devices (Continued)

The second output structure is actually an I/O pin, shown in Figure 9. The output logic is an active-low function of seven product terms. The data from the pin also feeds back into the AND array.

Notice that the output buffer is controlled by its own product term, allowing dynamic I/O control. This dynamic control can be used either to determine the ratio of device inputs to outputs or, to disable the outputs when connected in a bus environment.

Third is a sequential, or registered output, shown

in Figure 10. The logic OR of eight products terms is available to the designer. Here, the register state (both true and complement) feeds back into the array, as well as into an output buffer with a bank-controlled output enable feature. The clock is common as well, minimizing switching skew between buffers, and the register is a high speed D type. The feedback of the register data into the AND array allows the current state data (in the registers) to be part of the next state function. This is necessary for most sequential functions, such as counting and shifting operations.

Figure 9. Asynchronous I/O Structure

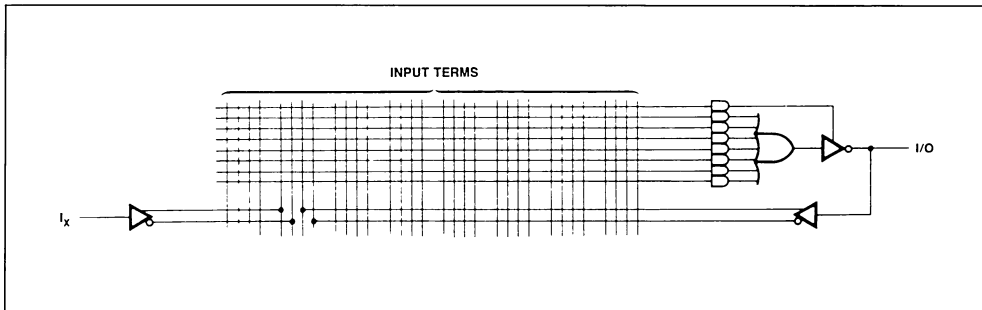
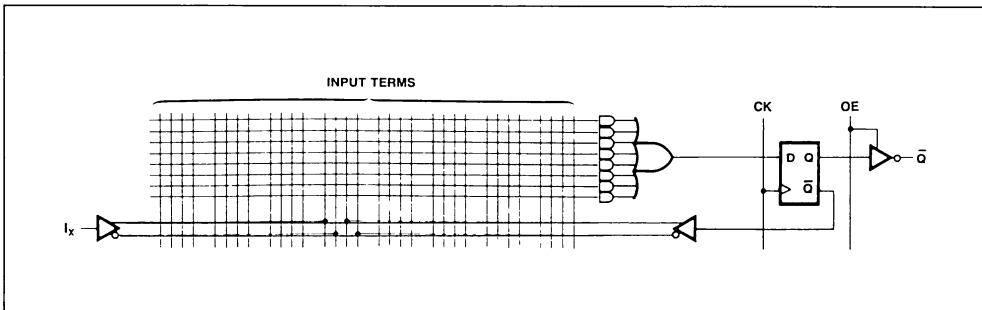


Figure 10. Sequential (Registered) Output Structure



GAL Devices

GAL devices have the same programmable AND array driving a fixed OR array. The difference is in the architecture and flexibility of the output functions.

The GAL device integrates an Output Logic Macrocell (OLMC) on each of its output pins. The GAL

16V8 and its eight OLMCs are shown in figure 14. The OLMCs, shown in detail in the figures 11 and 12, are configured by the designer function. Figure 13 shows the GAL macrocell configured into the four basic operating modes described previously.

Figure 11. GAL Device Output Logic Macrocell (OLMC): Pin 12 and 19 (GAL 16V8); Pin 15 and 22 (GAL 20V8)

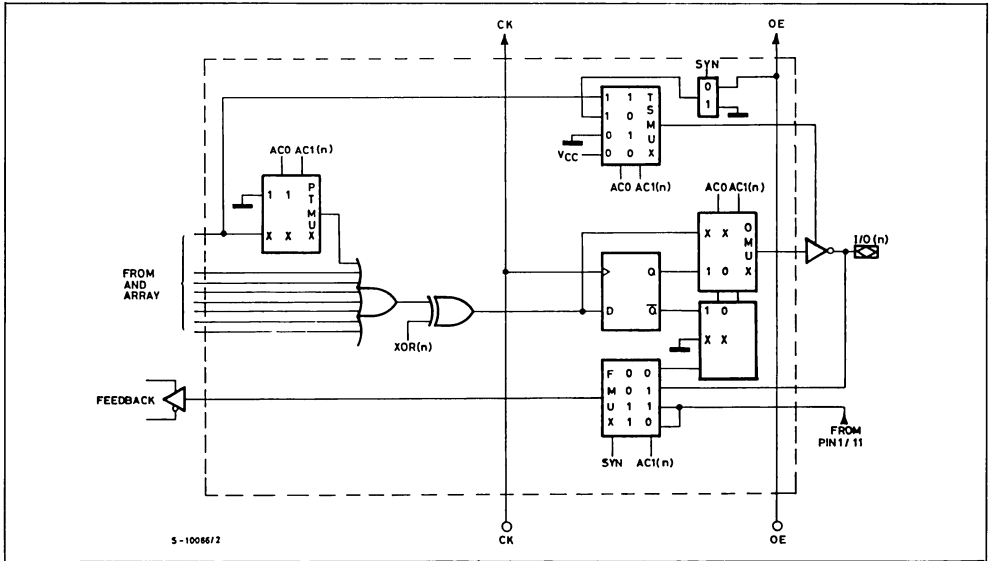


Figure 12. GAL Device Output Logic Macrocell (OLMC): Pin 13 to 18 (GAL 16V8); Pin 16 to 21 (GAL 20V8)

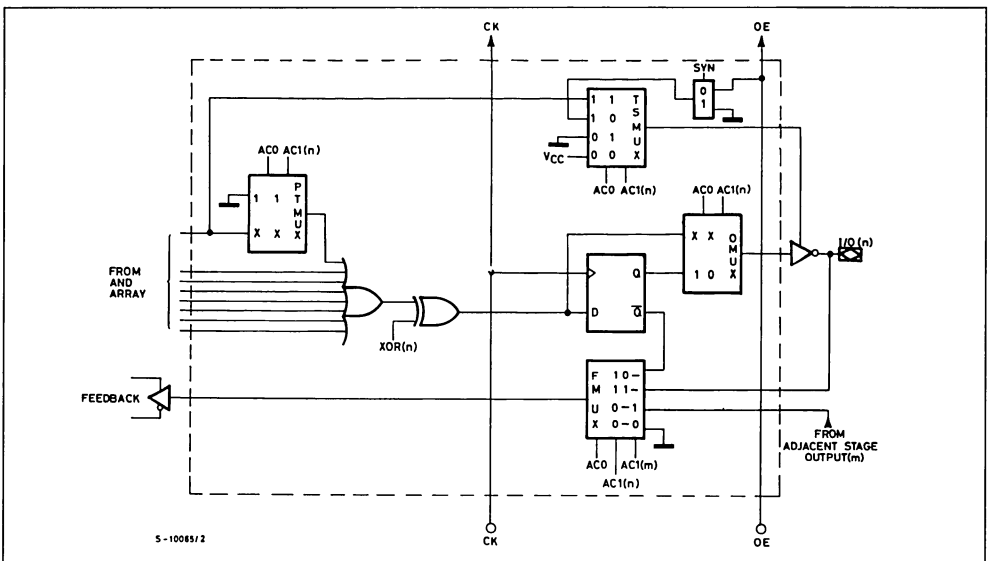
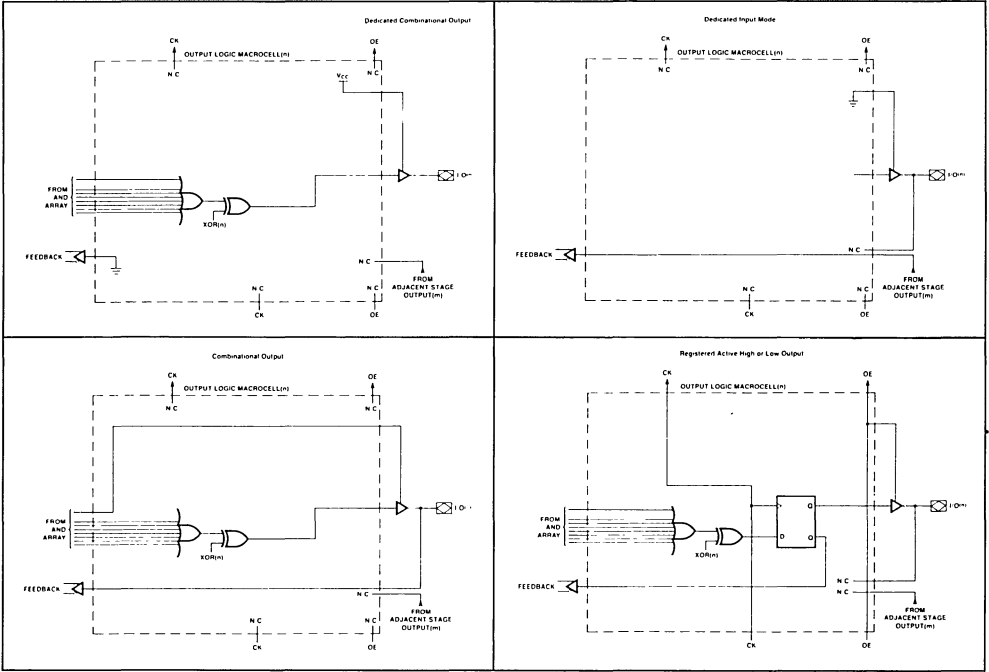


Figure 13. OLMC Configurations



The programmable polarity feature of each of the output macrocells deserves a special investigation. Located in the heart of the OLMC, the programmable polarity function is implemented by the exclusive-OR (XOR) gate that follows the OR gate from the array. Recalling the truth table of an XOR gate, it can be shown that the data can be inverted (control = 1) or not inverted (control = 0), depending on state of the second input, as shown in the table below.

Programmable Polarity Functionally

Data	Control	XOR Output
0	0	0
1	0	1
0	1	1
1	1	0

Programmable output polarity is used extensively in DeMorgan's Law to reduce the number of product terms required to implement a function. As a result, the GAL device can generally implement functions that appear to require more than 8 product terms per output. For example:

$$O = A + B + C + D + E + F + G + H + I$$

is a function of 9 product terms, each representing one input variable. This equation can be reduced to one product term if DeMorgan's Law is applied.

$$\bar{O} = \bar{A} * \bar{B} * \bar{C} * \bar{D} * \bar{E} * \bar{F} * \bar{G} * \bar{H} * \bar{I}$$

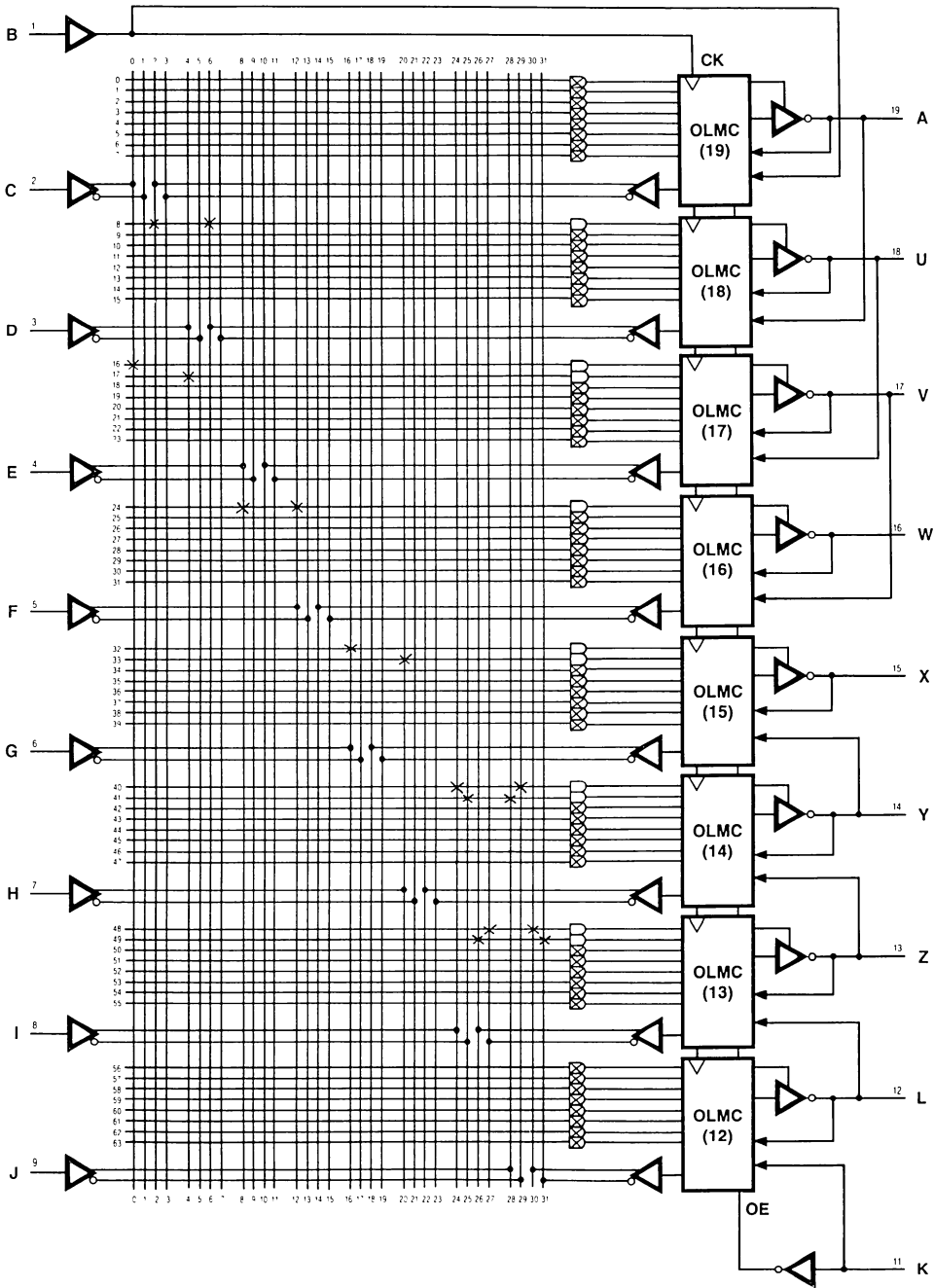
\bar{O} is now a function of only 1 product term. To obtain O again, we need only invert the function using the polarity feature of the GAL OLMC. This function could not have been implemented in the last generation PAL architecture.

The OLMCs are configurable by the designer to perform the various functions. For example, the designer merely specifies two active-low registers, one active-high register and the device is configured instantly.

Since each of the OLMCs contains the same logic, it is also possible to "tweak" an existing design for the convenience of the manufacturing department. One example might be moving a function to an adjacent pin to optimize board layout.

This could eliminate an interconnect level on a multilevel board, by swapping two functions and eliminating the need to cross traces. The GAL architecture is not fixed until the user specifies the requirements.

Figure 14. GAL 16V8 Block Diagram



GAL CONFIGURATION EXAMPLE

This example shows how the various GAL16V8 architecture bits are derived to implement a device with 12 inputs and 6 programmable combinatorial outputs (figure 15). In practice, the user does not be concerned with these architecture bits, since they are automatically handled by qualified programming equipment.

XOR(12) = X	Input pin, XOR not used
XOR(13) = 1	Invert
XOR(14) = 0	Non-invert
XOR(15) = 1	Invert
XOR(16) = 1	Invert
XOR(17) = 0	Non-invert
XOR(18) = 0	Non-invert
XOR(19) = X	Input pin, XOR not used

DERIVING THE ARCHITECTURE CONTROL WORD

The architecture control word bits for this example are determined as follows:

SYN = 1 All outputs are combinatorial (no registers)
 (Pins 1, 11 available as inputs in this configuration.)

AC0 = 0 No tri-state outputs

- AC1(12) = 1 Input
- AC1(13) = 0 Output
- AC1(14) = 0 Output
- AC1(15) = 0 Output
- AC1(16) = 0 Output
- AC1(17) = 0 Output
- AC1(18) = 0 Output
- AC1(19) = X Input

Since the above example matches a standard PAL architecture (PAL14P6), any logic assembler or compiler capable of supporting PAL14P6 could be used to develop this design.

The resulting PAL14P6 JEDEC file could then be downloaded into a suitable GAL programmer, which would automatically map the PAL14P6 fuses into the GAL16V8 cells, and set the GAL architecture bits accordingly.

Several logic compilers (ABEL, CUPL, PLAQ, LC9000) are capable of supporting GAL devices in non-PAL architectures. With these compilers, you can make a "GAL17P5", for example.

Figure 15

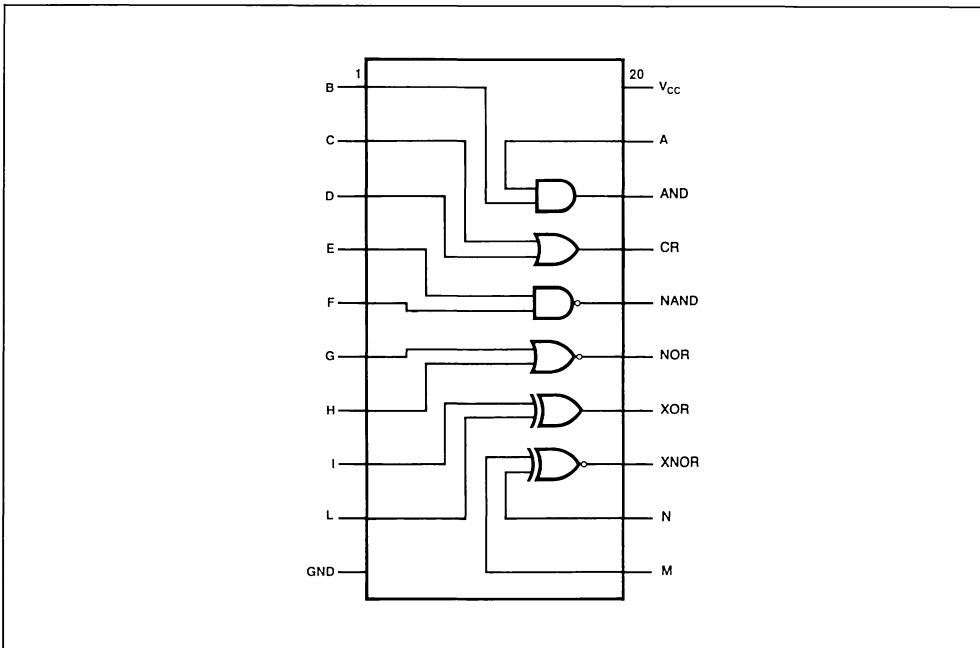
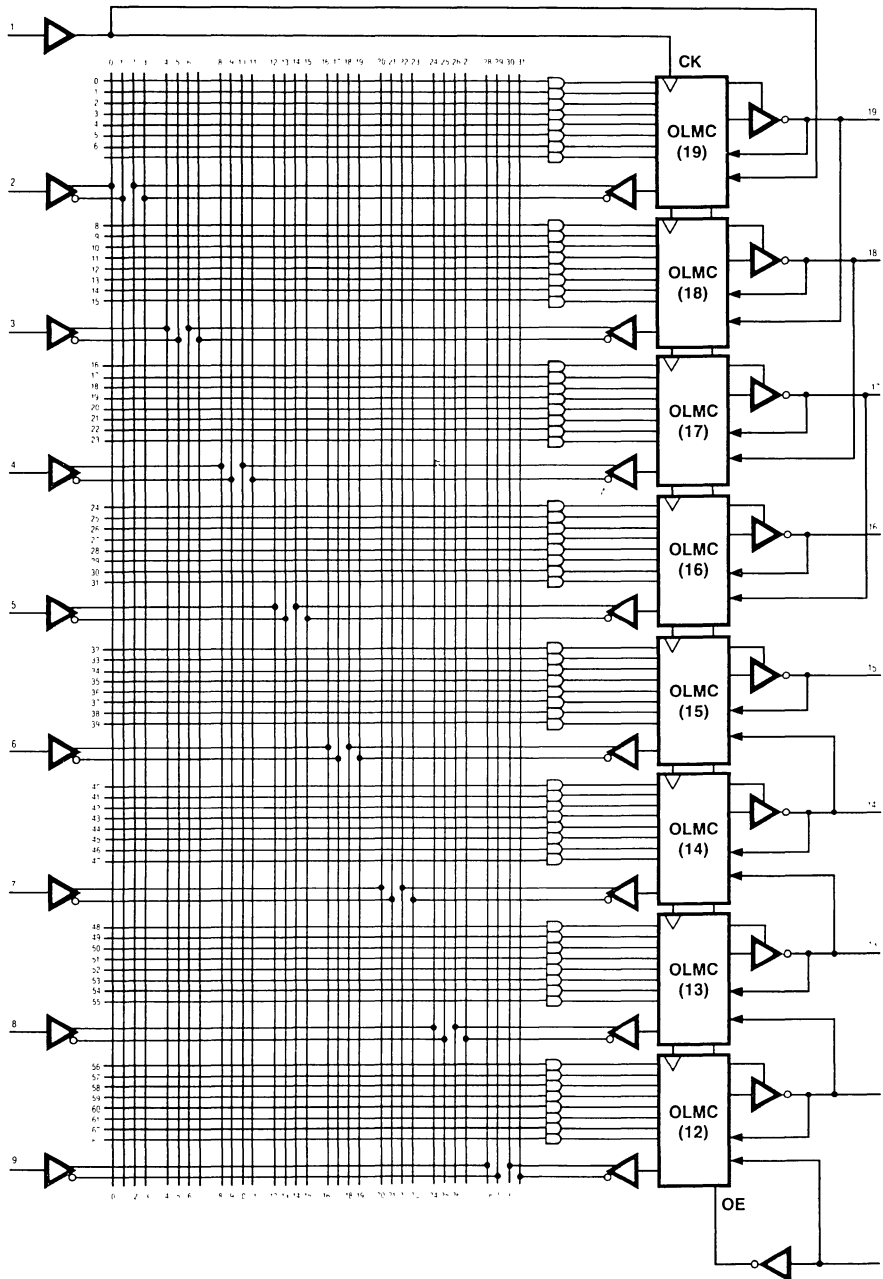


Figure 16. GAL 16V6 Block Diagram



Programming Basic Logic Gates

Examples of implementation of basic gates AND, OR and XNOR are shown. Programming other basic logic gates NAND, NOR and XOR can be obtained similarly.

AND

Logic Equation	Pins
$AND = A * B$	18,19,1

The cells A (row 2, column 8) and B (row 6, column 8) must be programmed. Figure 17 shows the equivalent electric circuit for the AND function.

The number in brackets indicates how many input product terms are programmed at the specified logic level

OR

Logic Equation	Pins
$OR = C + D$	17,2,3

XNOR

Logic Equation	Pins
$XNOR = N * \bar{M} + \bar{N} * M$	13,12,11

Also in this case two product terms are involved. We have to program four cells:

Cell	Row	Column
N	26	49
\bar{N}	27	48
M	30	48
\bar{M}	31	49

Figure 17

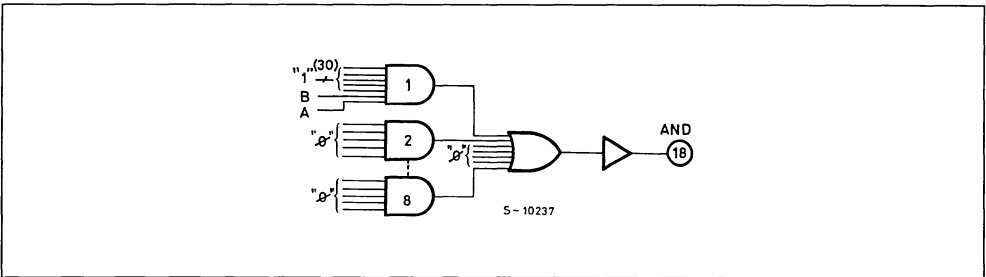
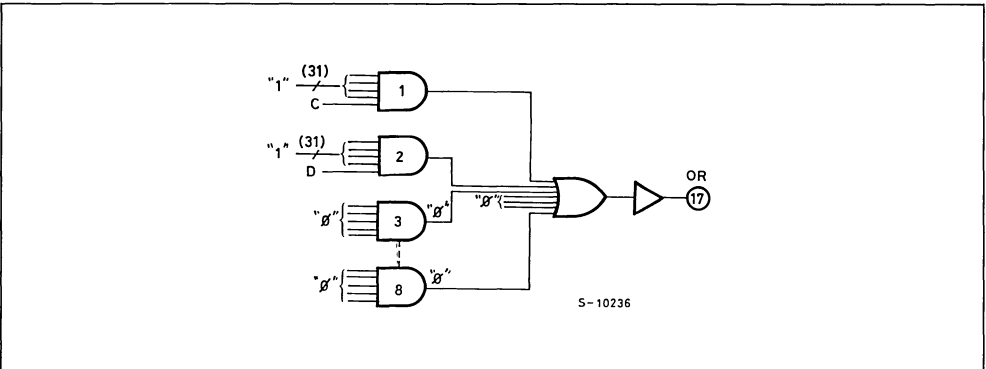


Figure 18



Programming Basic Logic Gates (Continued)

Figure 19

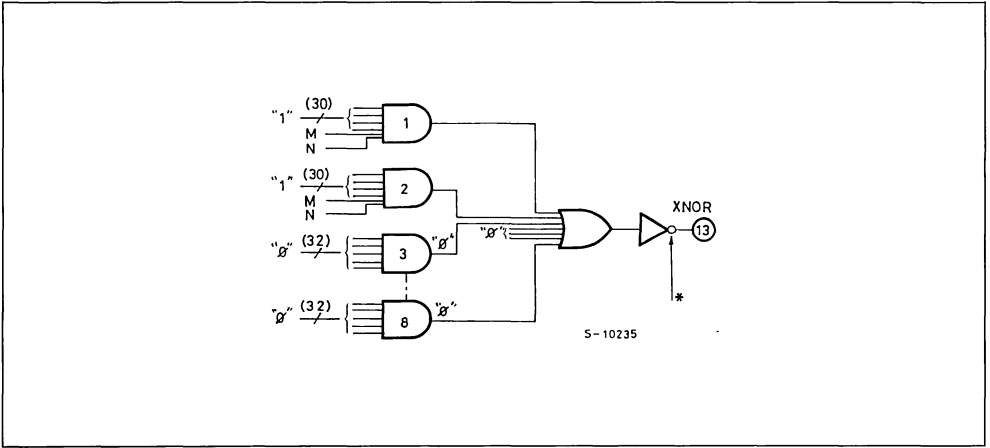


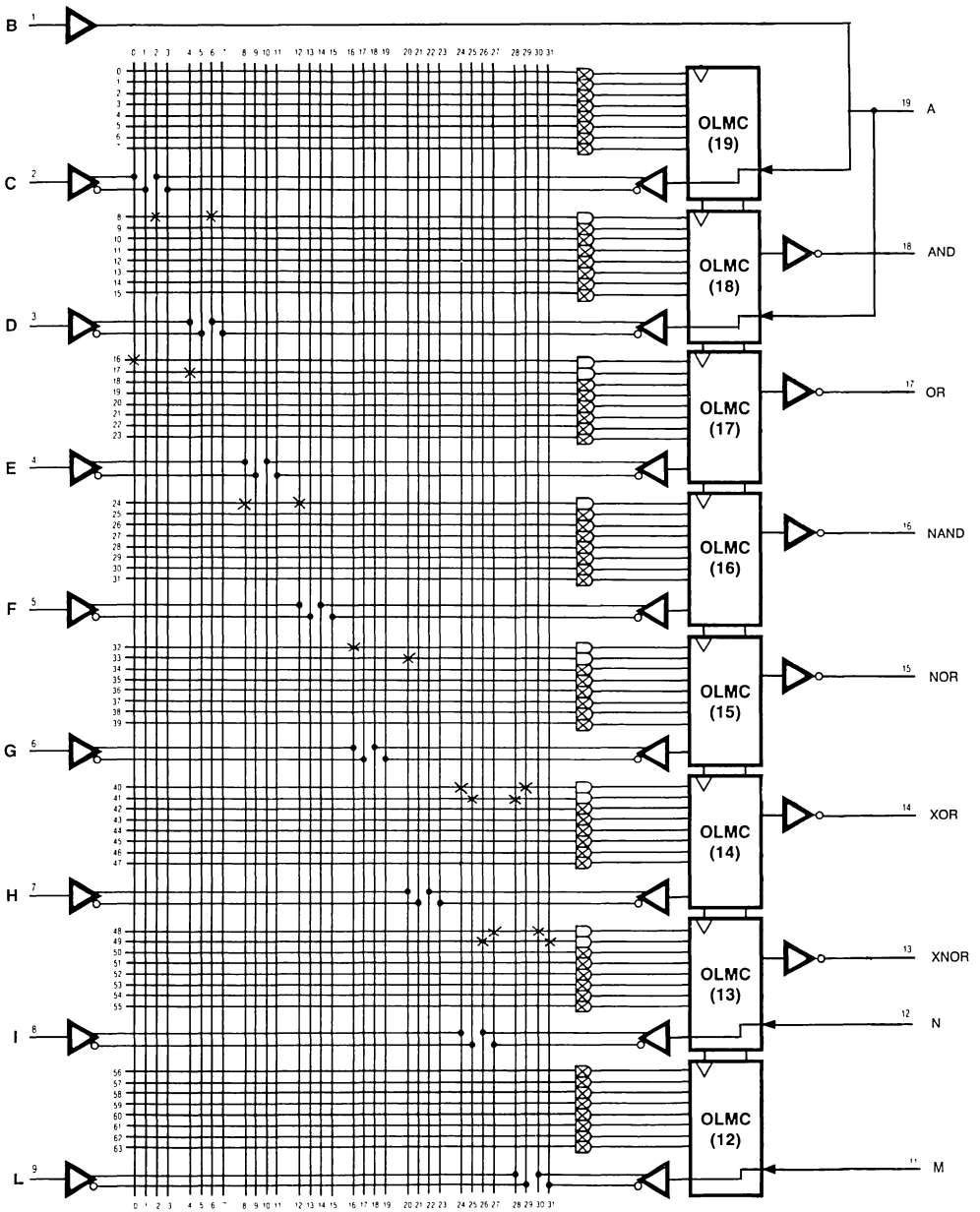
Figure 19 shows the equivalent circuit for the XNOR function.

* Inversion caused by polarity bit at logical "0"
Numbers in brackets indicate how many input product terms are programmed at the specified logical level.

PROGRAMMING RESULT

When a 16V8 has been programmed for the application shown in fig. 15, the logic diagram becomes that of figure 20.

Figure 20. GAL 16V8 Basic Gates Diagram



Fuse plot is shown on the following pages

FUSE PLOT FOR THE APPLICATION IN FIGURE 15

POLARITY	SYN/ACO	AC1
11111111		11111111
98765432	S/A	98765432
x - - x x - x x	- x	- x x x x x x -

11111111111222222222233 P
 01234567890123456789012345678901 D
 Pin #19 2048 Pol x 2120 Ac1 -

0000	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	Unused product terms must be programmed.
0032	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0064	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0096	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0128	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0160	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0190	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0224	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	

Pin #18 2049 Pol - 2121 Ac1 x

	B	A	
0256	--x--x-----		- AND : B & A;
0288	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	x Output is active-HIGH, because "POL" is not programmed (Pol -).
0320	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0352	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0384	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	Unused product terms must be programmed.
0416	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0448	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0480	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	

Pin #17 2050 Pol - 2122 Ac1 x

	C	D	
0512	x-----		- OR : C # D;
0544	--x-----		- Output is active-HIGH, because "Pol" is not programmed (Pol -).
0576	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0608	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0640	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	Unused product terms must be programmed.
0672	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0704	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0730	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	

Pin #16 2051 Pol x 2123 Ac1 x

	E	F	
0768	-----x-----		- NAND : $\overline{E} \& \overline{F}$;
0800	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	x Output is active-LOW because "Pol" is programmed (Pol x)
0832	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0864	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0896	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	Unused product terms must be programmed.
0928	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0960	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	
0992	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	

Legend x : Cell Programmed (logical "0")
 - : Cell Erased (logical "1")

FUSE PLOT FOR THE APPLICATION IN FIGURE 15 (cont'd)

POLARITY	SYN/ACO	AC1
11111111		11111111
98765432	S/A	98765432
x - - xx - xx	- x	- xxxxxx -

1111111111222222222233 P
 01234567890123456789012345678901 D

Pin # 15 2052 Pol x 2124 Ac1 x

		G	H	
1024	-----	x	-----	-
1056	-----	-	x	-----
1088	xx			x
1120	xx			x
1152	xx			x
1184	xx			x
1216	xx			x
1248	xx			x

NOR : $\overline{G \# H}$
 - Output is active-LOW, because "Pol" is programmed (Pol x)
 Unused product terms must be programmed.

Pin # 14 2053 Pol - 2125 Ac1 x

		I	\bar{I}	\bar{L}	
1280	-----	x	-----	x	-----
1312	-----	-	x	-----	-
1344	xx				x
1376	xx				x
1408	xx				x
1440	xx				x
1472	xx				x
1504	xx				x

XOR : $I \ \$ \ L;$
 - Logic operator XOR is equal to: $(I \ \& \ \bar{L}) \ \# \ (\bar{I} \ \& \ L)$
 - Output is active-HIGH because "Pol" is not programmed (Pol -)
 Unused product terms must be programmed.

Pin # 13 2054 Pol x 2126 Ac1 -

		\bar{N}	\bar{M}	\bar{M}	\bar{N}
1536	-----	x	-----	x	-----
1568	-----	-	x	-----	-
1600	xx				x
1632	xx				x
1664	xx				x
1696	xx				x
1728	xx				x
2760	xx				x

XNOR : $\overline{M \ \$ \ N};$
 - Output is active-LOW, because "Pol" is programmed (Pol x).
 Unused product terms must be programmed.

Pin # 12 2055 Pol x 2127 Ac1 -

1792	xx	x
1824	xx	x
1856	xx	x
1888	xx	x
1920	xx	x
1952	xx	x
1984	xx	x
2016	xx	x

Unused product terms must be programmed.

Legend x : Cell Programmed (logical "0")
 - : Cell Erased (logical "1")

DEVELOPMENT TOOLS

SOFTWARE AND HARDWARE TOOLS

SGS-THOMSON Microelectronics Inc. specializes in the business of designing and manufacturing EECMOS programmable logic devices as well as a broad range of commodity and semi-custom semiconductor components. As such SGS-THOMSON does not supply custom software or hardware to program our line of GAL devices. However many excellent third party suppliers provide software and hardware which fully support the SGS-THOMSON GAL devices as well as other PLD devices.

Most current vendors of PLD programming hardware and software support GAL devices. However, an upgrade of some programming hardware may be necessary to support GAL devices. Vendors and equipment model numbers which have been qualified on GAL devices are on file and may be obtained from SGS-THOMSON sales offices, a listing of which is provided at the back of this handbook.

A list of potential hardware vendors and the qualification status of each is provided in the appendix of this chapter. As this list will undergo constant updates as additional vendors are added, it may be instructive for the user to contact the local SGS-THOMSON sales office to determine if a specific vendor's hardware is qualified for programming GAL devices. SGS-THOMSON recommends that only qualified hardware be used to program GAL devices as this will ensure the 100% functional and programming yield of SGS-THOMSON GAL devices.

Software Tools

The availability of user-friendly and functional software tools is the main contribution to the tremendous upsurge in the usage of PLD devices. In the early 70's PLD's had a difficult time being accepted by systems and board designers due to the lack of good programming software. At that time it was necessary to load each individual fuse location into the devices, after extensive analysis of the design requirements. As this was a slow, cumbersome process requiring the designer to learn the architecture of many different devices in addition to the fact that logic errors could not be automatically identified, the acceptance of PLD's into mainstream system design was quite limited.

The late 70's saw the development of assembler software by various PLD vendors in order to help the sales of these devices. The most popular of these assemblers is PALASM*, from Monolithic Memories, which is today in the public domain and therefore not covered by software copyrights. This assembler allows inputs only as Boolean equations, has a difficult command structure, allows equations

only in a sum-of-products format, works on PAL devices only and has no intelligence i.e. unable to do logic minimization or identify specific device types which will or will not work with a given set of Boolean equations.

Of course the alternative, manual fuse patterning of the device, was much worse so assemblers found wide market acceptance for PAL device programming. The most severe restriction of this type of approach was the inflexibility of the software to work on other vendors devices thus forcing designers to learn many different assemblers in order to have more than one device supplier to choose from. Fortunately for PLD manufacturers and system designers a more generic software approach was not far away.

The development of compiler based software in the early 80's was a response to the need for more flexibility and utility in development tools. The original packages were developed by third-party manufacturers, not device vendors, with the goal of supporting all device types and all manufacturers. These original packages, CUPL* by Assisted Technology (now Personal CAD Systems of Los Gatos, CA) and ABEL* from Data I/O Corp. (Redmond, WA) had the capability of logic equation minimization, macros, truth table and state machine syntax and self-documentation (examples of the uses of these software packages are contained in this guide).

The latest advance in the PLD development software technology has occurred in the mid-'80s. These programs allow schematic capture using pre-programmed macros in the software which allow a designer to simply create a logic schematic as the input to a translator. The translator converts the graphic representation to a network list that is then compiled to the fuse maps by the software tool. All the other functions of the software such as logic minimization are then available to streamline the design before it is downloaded into a device. The most widely used of these tools are CAE-1 from Personal-CAD Systems (Los Gatos, CA) and DASH from FutureNet (Data I/O Corp., Redmond, WA).

Hardware Tools

The hardware used to program GAL devices (a list of qualified hardware vendors is available from SGS-THOMSON sales offices, listed at the back of this manual) can be divided roughly into two types; 1. the so-called "universal programmers" (in this case "universal" means with respect to PLD devices only, this terminology should not be confused with the broader sense of "universal" programmers, meaning those that program EPROM memories or EPROM

arrays in microprocessors as well as PLD's, although many of the programmers listed do program the above mentioned device families), and 2. the "GAL only" type programmers.

In the category of universal type programmers are those from Data I/O and Stag Microsystems as well as many others. These programmers support many different PLD devices, including ECL, CMOS EPROM, standard bipolar PALs and GAL EECMOS device types. These universal programmers also support many advanced functions such as test vectors, register preload and even chip handlers for automated handling in a production environment. The second type of programmers are the "GAL only" type programmers such as those from Qwerty and Programmable Logic Technologies. The Qwerty also supports test vectors and register preload for full functional testing of the GAL devices. The big advantage of the GAL only programmers is, of course, their low price which is from one-quarter to one-tenth as costly as the universal programmers. This makes them an excellent low-cost development system for GAL devices.

A further division can be made with respect to the operating mode of the programmers i.e. the "stand-alone" type units and those which require an IBM PC or equivalent to operate.

Generally the stand-alone units can read a device, store the JEDEC file in a memory and download the information to a newly loaded device including test vectors such that a volume "program and test" operation is possible without having a PC attached to each programmer.

Of course a connection to a peripheral such as a PC is still necessary to load a software developed JEDEC file into the programmer's memory.

The type of tool chosen should reflect the environment it will be used in i.e. a GAL-only programmer could be considered in an operation where GAL only development or a small volume of production is occurring in a situation where a low-cost evaluation and programming of GALs is necessary. However in a large development lab where many types of PLD's are being evaluated or a high volume production environment where automated handling of devices is necessary, a universal type programmer with chip handler may be more appropriate. A prime consideration should also be the necessary functions of the programmer as well. For example although SGS-THOMSON guarantees a 100% programming yield on our devices, test vector and register preload support capability is recommended for the designer to verify that the device is doing exactly what was planned.

This is especially important in state machine

design where bringing the outputs to a certain state simply by cycling the inputs can take an inordinate amount of time or may not even be possible.

With register preload the outputs can be driven to a specific state (one which may not occur in normal operation) and then next state operation can be checked to verify that the device returns to a normal, expected state. Where this can become an issue is after a power "glitch" in the system or after a power outage when the GAL outputs may be driven to an unknown or illegal state. Then when the power to the device returns to normal the GAL outputs may be in one of these indeterminate states.

Thus the concept of register pre-load allows the designer to test output conditions which may not normally occur but which the designer wishes to know anyway what the next state will be.

The designer can then verify that the device will return to a known, normal state under any conditions in its next state. This is just one example of the utility of the register pre-load function. Applying test vectors is simply the process of forcing specific inputs to the device and monitoring the outputs to verify the correct logical functioning of the device.

Programming the GAL is the process of providing it with the JEDEC file to perform a function and applying the specific series of voltage pulses to actually "burn" the JEDEC "fuse" map into the device. Also, if included in the JEDEC file, the programmer then applies the specific voltage pulses and looks at the outputs to run the test vector verification for functional testing of the device.

"Support" by a hardware manufacturer refers to his ability to provide the correct voltages and timing pulses and make the correct measurements on the outputs, if applicable, for the device. Given this the remainder of the process is merely downloading the JEDEC file to the GAL.

Downloading is the process of loading the JEDEC "fuse" map into the programmer. This "fuse" map can come from a pre-programmed device (i.e. master device), from a computer (a PC or mainframe with appropriate software to construct the JEDEC file from a number of logic or graphic entry methods), or from an attached peripheral such as a tape drive. If the file is transferred in JEDEC format (which is recommended by SGS-THOMSON), a checksum is calculated and verified at the end of data transmission to ensure that data was not lost or corrupted during transfer. Most programmers require a simple keystroke or pushbutton to put the hardware into the download mode.

The programming of the GAL is controlled by the programming hardware and therefore to ensure complete reliability of the SGS-THOMSON-GAL device SGS-THOMSON recommends that only approved programmers be used to program GAL devices. A list of qualified vendors is available from SGS-THOMSON upon request.

Since the GAL device is fabricated using an EEC-MOS technology it is erasable and reprogrammable. In fact the first step in the programming process is a bulk erase of the device. The actual number of program and erase cycles the device will tolerate is guaranteed by SGS-THOMSON to be at least 100. We have seen devices in our lab, however, which have thousands of cycles on them and still continue to function.

The ultimate test is a successful verification of the device which occurs automatically after programming. If no errors occur during the verify cycle the part is programmed and usable no matter how many program-erase cycles it has been through.

The patterning of the GAL device array is done using a parallel programming scheme. This allows the device to be programmed very fast and in fact is less than a second on most programming hardware. This is up to an order of magnitude faster than devices using the UV-CMOS approach. During this programming cycle, the logic array, the architecture matrix programming and the verify cycle are executed. The verify cycle checks programming and margins conservatively such that a minimum data retention of 20 years is ensured.

Finally the GAL device has a security cell which is programmable and erasable (but it can only be erased during a bulk erase when the entire array will also be erased). This prevents a "read" and therefore does not allow re-verification of the device.

This feature is provided to protect proprietary designs from competitive eyes. The fact that the security cell is erasable (simultaneously with the rest of the device) allowing the device to be reused, is a unique cost saving feature of EECMOS technology.

THE DESIGN PROCESS

By choosing generic, compiler-based software generic hardware and generic silicon (such as GAL devices), the biggest decision in the design process have already been made. The choice of the appropriate programmable logic device has traditionally been a difficult first step in starting a design, since with bipolar PLDs, you must guess which one of the dozens of architectures has the right com-

ination of outputs, I/Os and registers.

If your choice is wrong, you must guess again. The SGS-THOMSON GAL concept simplifies the approach, requiring that you merely count the number of inputs and outputs, then select a speed/power option. The development software automatically and dynamically allocates the inputs, I/Os, register, and so on.

The following design example is to implement the functions AND plus XNOR into a GAL16V8. The specific syntax is that of CUPL; however, other generic software (ABEL) has similar syntax and functions. In this cursory "walk-through", segments of code are presented as they would appear on the screen of a personal computer running CUPL software. The manufacturers of the software would, of course, be glad to provide a more comprehensive tutorial.

Once the software knows which device will be used, fields are provided for optional information, such as company name, design description, etc.

Tutorial Using a GAL16V8 Source File (401.PLD)

```
PARTNO    Y00;
NAME      XX1;
REV       1;
DATE      8/3/87;
DESIGNER   Alberto;
COMPANY   SGS-THOMSON;
ASSEMBLY  n/a;
LOCATION    n/a
```

The device pinout and pin labels need to be specified next. Use names that are convenient for you to reference, since the software doesn't care what you call a pin, as long as you are consistent:

```
/* inputs */
pin [1,2] = [A, !B];

/* outputs */
pin [18,19] = [Y, !Z];
```

It's a good idea to specify pin names in a format that is consistent with the actual pin state. In the above pin definitions, signals A,B and Y are active-high, while Z is active-low. We have chosen to indicate active-low data signals by prefixing the labels with exclamation points in the definition statement. The use of an active-high variation of these signals in subsequent design statements will automatically be resolved by the software compiler.

Entry of the logic functions is next. Traditionally, this entry is in the form of Boolean equations. Current revisions of development software allow truth-table, state-machine and schematic-entry formats, as well.

Here, the traditional equation-entry format is used to create an AND function on Y (pin 18) and an XOR function on Z (pin 19). Since Z has been defined as an active-low signal, however, we will acutally end up with XNOR on pin 19:

```
/* logic equations */
Y = A & B;
Z = !A & B # A & !B;
```

The operators used in the CUPL language are “!” for invert, “&” for the AND function and “#” for the OR function. The equations are written exactly as needed. All of the inversions for active-low inputs and outputs will be automatically resolved, a routine procedure for compiler software. Although these are simple equations, if they had been complex ones that needed automatic reduction to a specific number of product terms for a given PLD, the software would have performed that reduction, as well.

Next, the CUPL compiler needs to be invoked to process the “source” file, the text and equations provided above.

For the compiler to run one of its “modules” it must be told the target device type (“G16V8” in the example), source file name (“401”), and whatever additional functions the user would have it perform, through the use of flags (“-jfkfs”) that are passed at compile time.

The compile directive would typically appear as follows:

```
CUPL -jfkfs G16V8 401
CUPL Version 2.10B1 Copyright (c)
1983,84,85 Assisted Technology, Inc.
cuplx
time: 3 secs
cupla
time: 19 secs
cuplb
time: 7 secs
cuplm
time: 4 secs
cuplc
time: 15 secs
csima
time: 24 secs
total time: 72 secs
```

Some of the more recently announced PLD support products such as the Personal Silicon Foundry software available from DATA I/O use a very

friendly menu concept that saves defaults, eliminating the need to use flags.

The CUPL compiler produces a report, called a documentation file, part of which is presented here for reference. The purpose of the file is to provide a hard-copy documentation of the final (reduced) equations, the cell map or “fuse plot” and a chip-pinout diagram, if desired:

Device G16V8s Library DLIB-d-55-8
Created Mon Aug 3 06:30:10 1987

Expanded Product Terms

```
Y = > A & B
Z = > !A & B # !A & !B
```

Symbol Table

Pin Pol	Variable Name	Pin	Pterms Used	Max Pterms
	A	1	-	-
	B	2	-	-
	Y	18	1	8
!	Z	19	2	8

Fuse Plot

Syn 2192 - AC0 2193 x
Pin # 19 2048 Pol x 2120 Ac1 x

```
0000 -xx-----
0032 x--x-----
0064 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0096 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0128 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0160 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0192 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0224 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Pin # 18 2049 Pol - 2121 Ac1 x

```
0256 x-x-----
0288 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0320 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0352 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0384 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0416 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0448 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0480 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

(continued....)

Legend x : fuse not blown
- : fuse blown

THE DESIGN PROCESS (Cont'd)

Chip Diagram

```

*****
*
* 1      * 20 * VCC
*
* Y00
*
* 2      * 19 * !Z
*
*
* 3      * 18 * Y
*
*
* 4      * 17 *
*
*
* 5      * 16 *
*
*
* 6      * 15 *
*
*
* 7      * 14 *
*
*
* 8      * 13 *
*
* 9      * 12 *
*
*
* 10     * 11 *
*
*****

```

```

*QP20
*QF2194
*G0
*F0
*L0000 100111111111111111111111111111111111
*L0032 011011111111111111111111111111111111
*L0256 010111111111111111111111111111111111
*L2048 010000000000000000000000000000000000
*L2112 00000000001111111111111111111111111111
*L2144 1111111111111111111111111111111111111111
*L2176 11111111111111110
*C14D6

```

Test vectors, which indicate the stimulus and response for a PLD, serve primarily to validate the functionality of a desing source file. The CUPL compiler thus simulates the source file on paper so that, hopefully, only properly functioning patterns are ever programmed into a PLD for system debug. In our basic gates example, the source file for the simulator routine provides the expected data:

```

/* The Order Statement specifies the layout of
the function vector table.

%n = n spaces inserted between variables. */

order : A,B,%01,Y,%01,I,Z;
vectors:
/* AB Y Z */
00 L X /* test AND gate */
01 L X
10 L X
11 H X
00 X H /* test XNOR gate */
01 X L
10 X L
11 X H

```

The simulator processes the validation data, given above, compares it with that in the source file (using the original equations), to produce the following partial output:

Simulation Results

```

0001: 00 L N
0002: 01 L N
0003: 10 L N
0004: 11 H N
0005: 00 N H
0006: 01 N L
0007: 10 N L
0008: 11 N H

```

JEDEC, a standards organization with representatives from major semiconductor companies on its committes, has approved a standard for the interchange of PLD data. The JEDEC file is used as the medium of transfer from the development computer environment to that of the hardware device programmer. Included in the file are control bits that determine the status of security cell or fuses, test vectors, and data-transmission checksums. A portion of the JEDEC file for our example is reproduced here:

GAL DEVICE APPLICATIONS

INTRODUCTION TO GAL APPLICATIONS

The main advantage of the SGS-THOMSON GAL family is its high degree of freedom in the field of logic design.

With the SGS-THOMSON GAL family, the system designer is allowed to replace existing programmable logic; moreover he can use powerful logic compilers to optimize the design of new products. In this section five simple applications are discussed, they are: (see table)

Each application example is presented as a GAL design, complete with the required logic function and the actual GAL device logic implementation.

The GAL design specification and the actual CUPL and ABEL outputs for the programmer make the examples complete enough to serve as guides for those designers who are using SGS-THOMSON GALs in their own systems.

APPLICATION	GAL DEVICE	SOFTWARE
BASIC GATES	16V8	CUPL*/ABEL*
BASIC FLIP-FLOP	16V8	CUPL/ABEL
7 SEGMENT DISPLAY DRIVER	16V8	ABEL
QUAD 4:1 MULTIPLEXER	20V8	CUPL/ABEL
7 BIT COUNTER	20V8	CUPL/ABEL

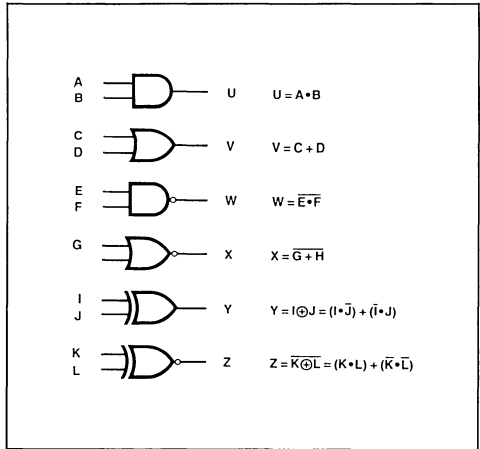
BASIC GATES

As a beginning example, the implementation of the basic gates: AND, OR, NAND, NOR, XOR, and XNOR (figure 1) in a GAL16V8 is shown. Since 12 inputs and 6 output are needed (figure 2), 2 Output Logic Macrocells (OLMCs) must be configured as dedicated inputs and 6 as dedicated combinational outputs. Programming software automatically handles this task. Active-high or low outputs are no problem either, because of the programmable polarity feature of the GAL16V8.

The program has been compiled using both CUPL and ABEL. Both programs need two files: the logic design input file (figures 3 and 11 respectively) and simulation file (figure 4 and 11).

For the design input file an equation must be written for each output, and pin assignments must be made. The simulation file contains "vectors" that test the functionality of the device. GAL devices are completely tested at the factory and post programming yields are guaranteed to be 100%; therefore, device testing is unnecessary. However since test vectors are useful for design verification, they are included here.

Figure 1. Basic Logic Gates



USING CUPL

A JEDEC file with test vectors is shown in figure 6. Because of the GAL device's fuse map compatibility with PAL devices, JEDEC files can be copied directly from PAL device applications into GAL devices. As shown in figure 7, 8 and 9, CUPL software will produce a documentation file which consists of:

- Expanded product terms (figure 7)
- Symbol table (figure 8)
- Chip diagram (figure 9)

The CUPL fuse plot is shown in figure 10

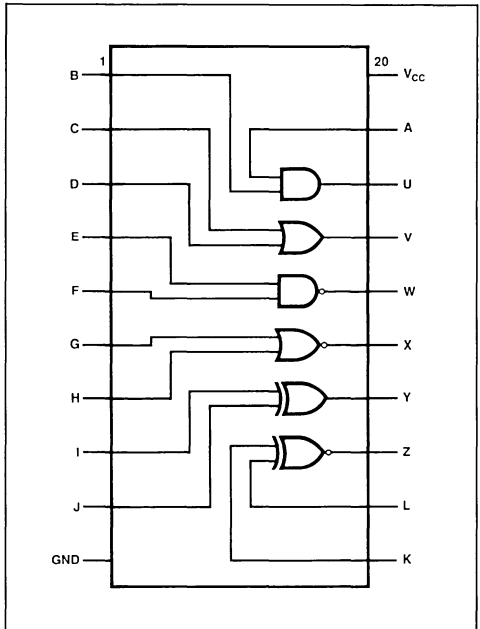
USING ABEL

ABEL software will produce a document generator file which consists of:

- Reduced equations (figure 12)
- Chip diagram (figure 13)
- Chip usage (figure 14)
- Test vectors (figure 15)

The JEDEC file, with test vectors and fuse map, is shown in figure 16.

Figure 2. GAL16V8 Basic Gates Pinout



BASIC GATES

Figure 3. CUPL Design Input File

```

Name      GATES;
Partno    00001;
Date      29/06/87;
Revision  01;
Designer  ALBERTO;
Company   SGS-THOMSON Microelectronics
Assembly  PC AT;
Location  PLZN1;
FORMAT    JEDEC;
/*****
/*
/* BASIC GATES : INPUT FILE
/*
/*
*****/
/* Allowable Target Device Types: G16V8
*****/

/** Inputs **/

Pin   [19,1] = [A,B] ; /* INPUTS AND */
Pin   [2,3] = [C,D] ; /* INPUTS OR */
Pin   [4,5] = [E,F] ; /* INPUTS NAND */
Pin   [6,7] = [G,H] ; /* INPUTS NOR */
Pin   [8,9] = [I,J] ; /* INPUTS XOR */
Pin   [11,12] = [K,L] ; /* INPUTS XNOR */

/** Outputs **/

Pin   18 = U ; /* OUTPUT AND */
Pin   17 = V ; /* OUTPUT OR */
Pin   16 = W ; /* OUTPUT NAND */
Pin   15 = X ; /* OUTPUT NOR */
Pin   14 = Y ; /* OUTPUT XOR */
Pin   13 = Z ; /* OUTPUT XNOR */

/** Declarations and Intermediate Variable Definitions **/

/** Logic Equations **/

U = A & B ; /* A N D */
V = C # D ; /* O R */
W = !(E & F) ; /* N A N D */
X = !(G # H) ; /* N O R */
Y = I $ J ; /* X O R */
Z = !(K $ L) ; /* X N O R */

```

BASIC GATES

Figure 4. CUPL Simulation File

CSIM Version 2.11b Serial# 5-00001-154
 Copyright (C) 1983,1986 Personal CAD Systems, Inc.
 CREATED Fri Sep 18 17:15:04 1987

LISTING FOR SIMULATION FILE: GATES.si

```

1: Name           GATES;
2: Partno        00001;
3: Revision      01;
4: Date          29/06/87;
5: Designer      ALBERTO;
6: Company       SGS-THOMSON Microelectronics
7: Location      PLZN1;
8: Assembly      PC AT;
9: Format         JEDEC;
10: /******
11: /*
12: /* BASIC GATES : SIMULATION FILE
13: /*
14: /******
15: /* Target Devices: G16V8
16: /******
17:
18:
19: Order: A,%2,U,%3,C,D,%2,V,%3,E,F,%2,W,%3,G,H,%2,X,
        %3,I,J,%2,Y,%3,K,L,%2,Z;
20:
21:
    
```

=====
 Simulation Results
 =====

	A	N	O	R	D	H	R	I	J	K	L	
	N	A	O	R	D	H	R	I	J	K	L	
	AB	D	CD	R	EF	D	GH	R	IJ	R	KL	R
0001:	00	L	00	L	00	H	00	H	00	L	00	H
0002:	10	L	10	H	10	H	10	L	10	H	10	L
0003:	01	L	01	H	01	H	01	L	01	H	01	L
0004:	11	H	11	H	11	L	11	L	11	L	11	H

BASIC GATES

Figure 6. CUPL JEDEC File with Test Vectors

```

CUPL                2.11b  Serial# 5-00001-154
Device              g16v8s  Library DLIB-f-23-8
Created             Fri Sep 18 17:07:25 1987
Name                GATES
Partno              00001
Revision            01
Date                29/06/87
Designer            ALBERTO
Company             SGS-THOMSON Microelectronics
Assembly            PC AT
Location            PLZN1
*QP20
*QF2194
*QV4
*GO
*FO
*L0256 11011101111111111111111111111111
*L0512 01111111111111111111111111111111
*L0544 11110111111111111111111111111111
*L0768 11111111011101111111111111111111
*L1024 11111111111111110111111111111111
*L1056 11111111111111111111110111111111
*L1280 1111111111111111111111101111011
*L1312 11111111111111111111111110110111
*L1536 1111111111111111111111111101101
*L1568 1111111111111111111111111011110
*L2048 01100100000000000000000000000000
*L2112 00000000100000011111111111111111
*L2144 11111111111111111111111111111111
*L2176 111111111111111110
*C2F02
*P 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*V0001 00000000N00HLLHLLON
*V0002 01010101GN10LHLHLL1N
*V0003 101010101N01LHLHLLON
*V0004 111111111N11HLLHLL1N
*DF00
    
```

BASIC GATES

Figure 7. CUPL Expanded Product Terms

```

*****
                                  GATES
*****
CUPL          2.11b Serial# 5-00001-154
Device        g16v8s Library DLIB-f-23-8
Created       Fri Sep 18 17:07:24 1987
Name          GATES
Partno        00001
Revision      01
Date          29/06/87
Designer      ALBERTO
Company       SGS-THOMSON Microelectronics
Assembly      PC AT
Location      PLZN1

```

```

=====
                          Expanded Product Terms
=====

```

```

U =>
    A & B

V =>
    C
    # D

W =>
    E & F

X =>
    G
    # H

Y =>
    I & !J
    # !I & J

Z =>
    K & !L
    # !K & L

```

BASIC GATES

Figure 8. CUPL Symbol Table

```

=====
Symbol Table
=====

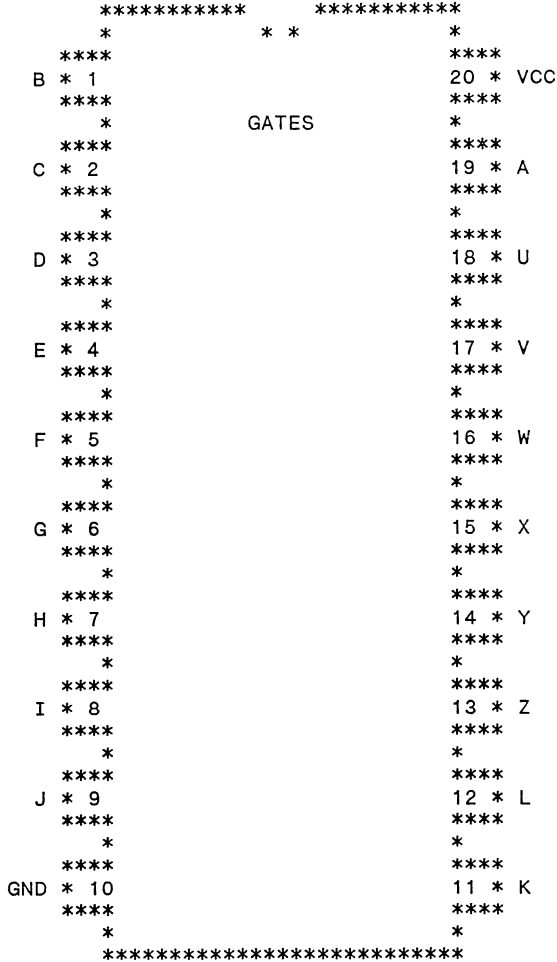
```

Pin	Variable	Ext	Pin	Type	Pterms	Max	Min
Pol	Name				Used	Pterms	Level
---	-----	---	---	---	-----	-----	-----
	A		19	V	-	-	-
	B		1	V	-	-	-
	C		2	V	-	-	-
	D		3	V	-	-	-
	E		4	V	-	-	-
	F		5	V	-	-	-
	G		6	V	-	-	-
	H		7	V	-	-	-
	I		8	V	-	-	-
	J		9	V	-	-	-
	K		11	V	-	-	-
	L		12	V	-	-	-
	U		18	V	1	8	1
	V		17	V	2	8	1
	W		16	V	1	8	1
	X		15	V	2	8	1
	Y		14	V	2	8	1
	Z		13	V	2	8	1

LEGEND F : field D : default variable M : extended node
 N : node I : intermediate variable T : function
 V : variable X : extended variable U : undefined

BASIC GATES

Figure 9. CUPL Chip Diagram



BASIC GATES

Figure 10. CUPL Fuse Plot

Syn 2192 - Ac0 2193 x

```

Pin #19 2048 Po1 x 2120 Ac1 -
0000 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0032 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0064 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0096 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0128 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0160 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0192 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0224 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #18 2049 Po1 - 2121 Ac1 x
0256 --x---x-----
0288 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0320 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0352 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0384 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0416 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0448 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0480 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #17 2050 Po1 - 2122 Ac1 x
0512 x-----
0544 ----x-----
0576 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0608 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0640 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0672 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0704 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0736 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #16 2051 Po1 x 2123 Ac1 x
0768 -----x---x-----
0800 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0832 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0864 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0896 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0928 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0960 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0992 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #15 2052 Po1 x 2124 Ac1 x
1024 -----x-----
1056 -----x-----
1088 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1120 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1152 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1184 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1216 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1248 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #14 2053 Po1 - 2125 Ac1 x
1280 -----x---x---
1312 -----x---x---
1344 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1376 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1408 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1440 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1472 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1504 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #13 2054 Po1 x 2126 Ac1 x
1536 -----x---x---
1568 -----x---x---
1600 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1632 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1664 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1696 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1728 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1760 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #12 2055 Po1 x 2127 Ac1 -
1792 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1824 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1856 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1888 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1920 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1952 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1984 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
2016 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```

LEGEND X : fuse not blown
 - : fuse blown

BASIC GATES

Figure 11. ABEL Input File and Simulation File

```

module  BASGATE
title  ' Basic Gates example
      By Derrick Tuten 13 July 1987
      SGS-THOMSON Microelectronics

" device declaration

" location keyword  device code
  U001  device      'P16V8S';

" pin declaration

" inputs
  A,B,C,D,E,F  pin 19,1,2,3,4,5;
  G,H,I,J,K,L  pin 6,7,8,9,11,12;

" outputs
  U,V,W,X,Y,Z  pin 13,14,15,16,17,18;

equations
  Z = A & B;      " And Gate

  !X = E & F;     " Nand Gate

  Y = C # D;     " Or Gate

  !W = G # H;    " Nor Gate

  V = I $ J;     " Exclusive Or Gate

  U = K !$ L;    " Exclusive Nor Gate

test_vectors  'Test And Gate'
  ( [A,B] -> Z )
  [0,0] -> 0;
  [0,1] -> 0;
  [1,0] -> 0;
  [1,1] -> 1;

test_vectors  'Test Nand Gate'
  ( [E,F] -> X )
  [0,0] -> 1;
  [0,1] -> 1;
  [1,0] -> 1;
  [1,1] -> 0;

test_vectors  'Test Or Gate'
  ( [C,D] -> Y )
  [0,0] -> 0;
  [0,1] -> 1;
  [1,0] -> 1;
  [1,1] -> 1;

```

BASIC GATES

Figure 11. ABEL Input File and Simulation File (Cont'd)

```

test_vectors 'Test Nor Gate'
  ( [G,H] -> W )
    [0,0] -> 1;
    [0,1] -> 0;
    [1,0] -> 0;
    [1,1] -> 0;

test_vectors 'Test Exclusive Or Gate'
  ( [I,J] -> V )
    [0,0] -> 0;
    [0,1] -> 1;
    [1,0] -> 1;
    [1,1] -> 0;

test_vectors 'Test Exclusive Nor Gate'
  ( [K,L] -> U )
    [0,0] -> 1;
    [0,1] -> 0;
    [1,0] -> 0;
    [1,1] -> 1;
end BASGATE

```

ABEL(tm) Version 2.10a - Document Generator
 Basic Gates example
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 Equations for Module BASGATE

Device U001

- Reduced Equations:

```

Z = (A & B);
X = !(E & F);
Y = (C # D);
W = !(G # H);
V = (!I & J # I & !J);
U = (!K & !L # K & L);

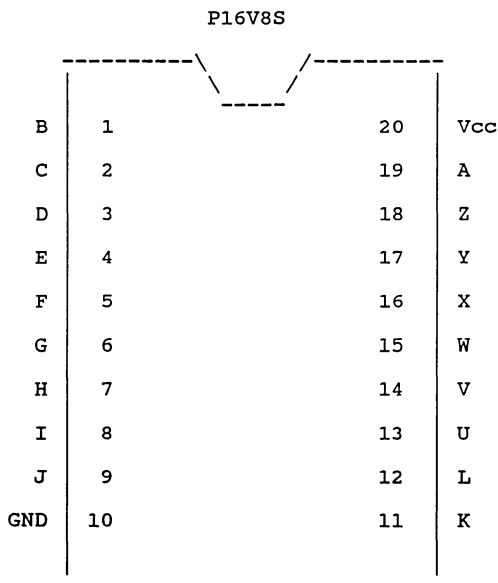
```


BASIC GATES

Figure 13. Chip Diagram

ABEL(tm) Version 2.10a - Document Generator
Basic Gates example
By Derrick Tuten 13 July 1987
SGS-THOMSON Microelectronics
Chip diagram for Module BASGATE

Device U001



BASIC GATES

Figure 14. Chip Usage

ABEL(tm) Version 2.10a' - Document Generator
 Basic Gates example
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 for Module BASGATE

Device U001

Device Type: P16V8S

Terms Used: 10 out of 64

Pin #	Name	Terms			Term Type	Pin Type
		Used	Max			
1	B	--	--	---	Input	
2	C	--	--	---	Input	
3	D	--	--	---	Input	
4	E	--	--	---	Input	
5	F	--	--	---	Input	
6	G	--	--	---	Input	
7	H	--	--	---	Input	
8	I	--	--	---	Input	
9	J	--	--	---	Input	
10	GND	--	--	---	GND	
11	K	--	--	---	Input	
12	L	0	8	Normal	I/O	
13	U	2	8	Normal	I/O	
14	V	2	8	Normal	I/O	
15	W	2	8	Normal	Output	
16	X	1	8	Normal	Output	
17	Y	2	8	Normal	I/O	
18	Z	1	8	Normal	I/O	
19	A	0	8	Normal	I/O	
20	Vcc	--	--	---	VCC	

BASIC GATES

Figure 15. Test Vectors

ABEL(tm) Version 2.10a - Document Generator
 Basic Gates example
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 Test Vectors for Module BASGATE

Device U001

Test And Gate

```

1 [0--- ---- ---- ---- --0-] -> [---- ---- ---- ---- -L--];
2 [1--- ---- ---- ---- --0-] -> [---- ---- ---- ---- -L--];
3 [0--- ---- ---- ---- --1-] -> [---- ---- ---- ---- -L--];
4 [1--- ---- ---- ---- --1-] -> [---- ---- ---- ---- -H--];
    
```

Test Nand Gate

```

5 [---0 0--- ---- ---- ----] -> [---- ---- ---- ---H ----];
6 [---0 1--- ---- ---- ----] -> [---- ---- ---- ---H ----];
7 [---1 0--- ---- ---- ----] -> [---- ---- ---- ---H ----];
8 [---1 1--- ---- ---- ----] -> [---- ---- ---- ---L ----];
    
```

Test Or Gate

```

9 [-00- ---- ---- ---- ----] -> [---- ---- ---- ---L ----];
10 [-01- ---- ---- ---- ----] -> [---- ---- ---- ---H ----];
11 [-10- ---- ---- ---- ----] -> [---- ---- ---- ---H ----];
12 [-11- ---- ---- ---- ----] -> [---- ---- ---- ---H ----];
    
```

Test Nor Gate

```

13 [---- -00- ---- ---- ----] -> [---- ---- ---- --H- ----];
14 [---- -01- ---- ---- ----] -> [---- ---- ---- --L- ----];
15 [---- -10- ---- ---- ----] -> [---- ---- ---- --L- ----];
16 [---- -11- ---- ---- ----] -> [---- ---- ---- --L- ----];
    
```

Test Exclusive Or Gate

```

17 [---- ---0 0--- ---- ----] -> [---- ---- ---- -L-- ----];
18 [---- ---0 1--- ---- ----] -> [---- ---- ---- -H-- ----];
19 [---- ---1 0--- ---- ----] -> [---- ---- ---- -H-- ----];
20 [---- ---1 1--- ---- ----] -> [---- ---- ---- -L-- ----];
    
```

Test Exclusive Nor Gate

```

21 [---- ---- --00 ---- ----] -> [---- ---- ---- H--- ----];
22 [---- ---- --01 ---- ----] -> [---- ---- ---- L--- ----];
23 [---- ---- --10 ---- ----] -> [---- ---- ---- L--- ----];
24 [---- ---- --11 ---- ----] -> [---- ---- ---- H--- ----];
    
```

end of module BASGATE

BASIC GATES

Figure 16. JEDEC File with Test Vectors (Cont'd)

```

00000000000000000000000000000000
1111111111111111111111111111101110
1111111111111111111111111111011101
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000*
L2048
01100110*
L2056
00000000000000000000000000000000000000000000000000000000000000000000*
L2120
1*
L2121
0*
L2122
0*
L2123
0*
L2124
0*
L2125
0*
L2126
0*
L2127
1*
L2128
11111111111111111111111111111111111111111111111111111111111111111111*
L2192
10*
N Test And Gate*
V0001 OXXXXXXXXXXXXXXXXXLON*
V0002 LXXXXXXXXXXXXXXXXXLON*
V0003 OXXXXXXXXXXXXXXXXXL1N*
V0004 LXXXXXXXXXXXXXXXXXH1N*
N Test Nand Gate*
V0005 XXX00XXXXNXXXXNHXXXXN*
V0006 XXX01XXXXNXXXXNHXXXXN*
V0007 XXX10XXXXNXXXXNHXXXXN*
V0008 XXX11XXXXNXXXXNLXXXXN*
N Test Or Gate*
V0009 X00XXXXXXXXXXXXXXXXNLXXN*
V0010 X01XXXXXXXXXXXXXXXXNHXXN*
V0011 X10XXXXXXXXXXXXXXXXNHXXN*
V0012 X11XXXXXXXXXXXXXXXXNHXXN*

```

BASIC GATES

Figure 16. JEDEC File with Test Vectors (Cont'd)

```
N Test Nor Gate*
V0013 XXXXX00XXNXXXHNXXXN*
V0014 XXXXX01XXNXXXLNXXXN*
V0015 XXXXX10XXNXXXLNXXXN*
V0016 XXXXX11XXNXXXLNXXXN*
N Test Exclusive Or Gate*
V0017 XXXXXX00NXXXLNXXXN*
V0018 XXXXXX01NXXXHNXXXN*
V0019 XXXXXX10NXXXHNXXXN*
V0020 XXXXXX11NXXXLNXXXN*
N Test Exclusive Nor Gate*
V0021 XXXXXXXXN00HXNXXXN*
V0022 XXXXXXXXN01LXNXXXN*
V0023 XXXXXXXXN10LXNXXXN*
V0024 XXXXXXXXN11HXNXXXN*
C2F42*
D9EA
```

BASIC FLIP-FLOP

In the Basic Gates application on the preceding pages, each "gate" was directly connected to an output pin. Here, the output registers of the GAL16V8 are used. A simple RS latch, a T (Toggle) flip-flop, a D flip-flop, and JK flip-flop are incorporated into a GAL16V8 (figure 1). Each is shown with its truth table and defining equations in figures 2 to 5. Note that all 3 flip-flops have synchronous preset (PR) and clear (CLR) inputs, while the RS latch does not. Also, the RS latch is not connected to the clock input; this was done to show the versatility of the GAL16V8; with a GAL device, the user is not locked into a specific architecture.

Design input file and simulation file both for CUPL and ABEL are constructed in a manner similar to the Basic Gates example. Each output must be given a distinct name, and any clocked circuit must be denoted with an appropriate extension (D) in the logic equations. The simulation file is again provided for design verification.

This example has some subtle requirements that may not be apparent to the first time user. When the RS latch is not being tested, it must remain in its latched state with output levels specified, or with the variable N (not tested) in their place. Also, when executing a preset or clear, remember that it will affect all flip-flops; even those not being tested will still respond.

Finally, all output levels should be specified or marked with the variable N; the variable X, which indicates a "don't care" condition, will not suffice. The device was programmed using both CUPL and ABEL.

Figure 1. Basic Flip-Flops Pinout

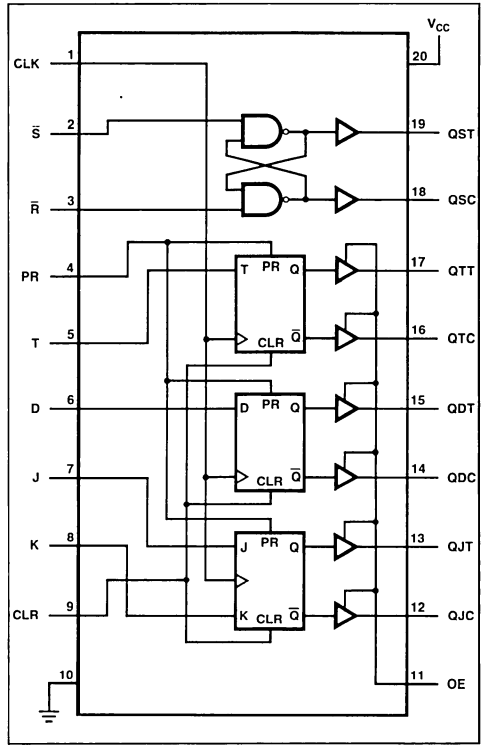


Figure 2. RS Latch

S	R	Q _N	Q _{N+1}	Q̄ _{N+1}	COMMENTS
0	0	0	1	1	Invalid
0	0	1	1	1	
0	1	0	1	0	Set
0	1	1	1	0	
1	0	0	0	1	Reset
1	0	1	0	1	
1	1	0	0	1	Latch
1	1	1	1	0	

BASIC FLIP-FLOP

Figure 3. T Flip-Flop

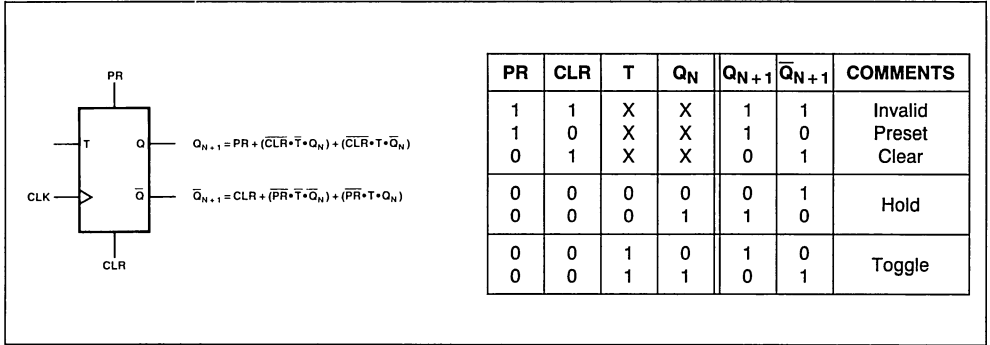


Figure 4. D Flip-Flop

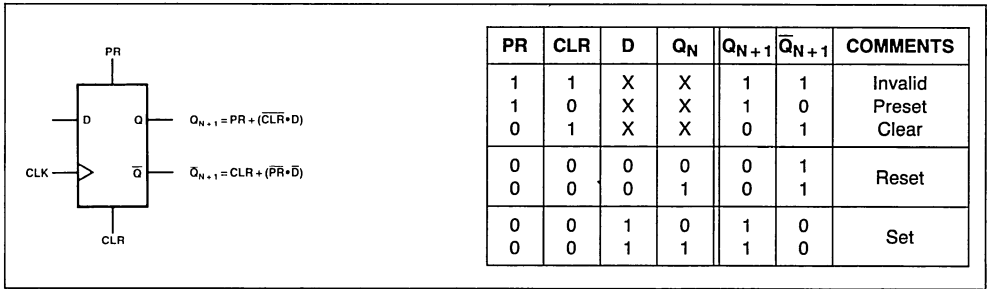
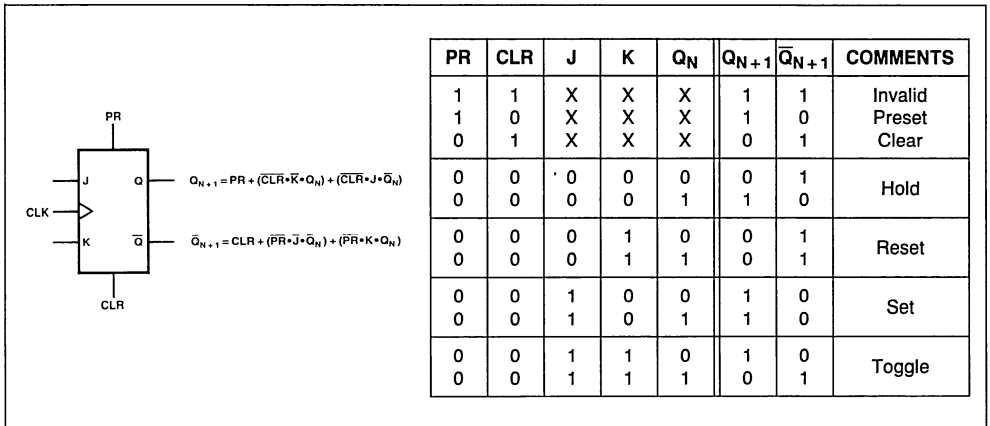


Figure 5. JK Flip-Flop



BASIC FLIP-FLOP

USING CUPL

The JEDEC file for this particular application is shown in figure 8 and 9. All the elements of the documentation file are shown in figures 10, 11 and 12. The CUPL "fuse" plot is shown in figure 13.

USING ABEL

The document generator file is shown in figures 15 through 18. The JEDEC file with fuse map and test vectors is shown in figure 19.

Figure 6. CUPL Input File

```
Name          FLIPFLOP;
Partno        00001;
Date          21/05/87;
Revision      01;
Designer      ALBERTO;
Company       SGS-THOMSON Microelectronics
Assembly     PC AT;
Location      PLZN1;
FORMAT        JEDEC;
/*****
/*
/* BASIC FLIPFLOP      :      INPUT FILE
/*
/*****
/* Allowable Target Device Types: G16V8
/*****

/** Inputs **/

Pin    [2,3] = [S,R]      ;      /* INPUT FLIPFLOP S-R      */
Pin    5 = T              ;      /* INPUT FLIPFLOP T        */
Pin    6 = D              ;      /* INPUT FLIPFLOP D        */
Pin    [7,8] = [J,K]     ;      /* INPUT FLIPFLOP J-K     */
Pin    [4,9] = [PR,CLR]  ;      /* PRESET AND RESET       */
Pin    [1,11] = [CLK,OE] ;      /* CLOCK AND OUTPUT ENABLE */

/** Outputs **/

Pin    [19,18] = [QST,QSC] ;      /* OUTPUT FLIPFLOP S-R    */
Pin    [17,16] = [QTT,QTC] ;      /* OUTPUT FLIPFLOP T      */
Pin    [15,14] = [QDT,QDC] ;      /* OUTPUT FLIPFLOP D      */
Pin    [13,12] = [QJT,QJC] ;      /* OUTPUT FLIPFLOP J-K    */

/** Declarations and Intermediate Variable Definitions **/

/** Logic Equations **/

QST = !S # (R & QST) ;
QSC = !R # (S & QSC) ;
QTT.D = PR # (!CLR & !T & QTT) # (!CLR & T & QTC) ;
QTC.D = CLR # (!PR & !T & QTC) # (!PR & T & QTT) ;
QDT.D = PR # (!CLR & D) ;
QDC.D = CLR # (!D & !PR) ;
QJT.D = PR # (!CLR & !K & QJT) # (!CLR & J & QJC) ;
QJC.D = CLR # (!PR & !J & QJC) # (!PR & K & QJT) ;
```

BASIC FLIP-FLOP

Figure 7. CUPL Simulation File

CSIM Version 2.11b Serial# 5-00001-154
 Copyright (C) 1983,1986 Personal CAD Systems, Inc.
 CREATED Fri Sep 18 17:58:21 1987

LISTING FOR SIMULATION FILE: FLIPFLOP.si

```

1: Name          FLIPFLOP;
2: Partno       00001;
3: Revision     01;
4: Date        21/05/87;
5: Designer     ALBERTO;
6: Company     SGS-THOMSON Microelectronics
7: Location    PLZN1;
8: Assembly    PC AT;
9: Format       JEDEC;
10: /*****
11: /*
12: /* BASIC FLIPFLOP : SIMULATION FILE
13: /*
14: /*****
15: /* Target Devices: G16V8
16: /*****
17:
18:
19: Order: OE,%3,CLK,%3,S,R,%2,QST,%2,QSC,%3,PR,%3,CLR,%3,T,%2,
20:      QTT,%2,QTC,%3,D,%2,QDT,%2,QDC,%2,J,K,%2,QJT,%2,QJC;
21:
    
```

Simulation Results

	OE	CLK	SR	QST	QSC	PR	CLR	T	QTT	QTC	D	QDT	QDC	JK	QJT	QJC
0001:	X	X	10	L	H	X	X	X	N	N	X	N	N	XX	N	N
0002:	X	X	01	H	L	X	X	X	N	N	X	N	N	XX	N	N
0003:	X	X	11	H	L	X	X	X	N	N	X	N	N	XX	N	N
0004:	X	X	00	H	H	X	X	X	N	N	X	N	N	XX	N	N
0005:	X	X	11	H	H	X	X	X	N	N	X	N	N	XX	N	N
0006:	0	C	11	N	N	1	0	X	H	L	X	H	L	XX	H	L
0007:	0	C	11	N	N	0	0	1	L	H	X	N	N	XX	N	N
0008:	0	C	11	N	N	0	0	1	H	L	X	N	N	XX	N	N
0009:	0	C	11	N	N	0	0	0	H	L	X	N	N	XX	N	N
0010:	1	C	11	N	N	0	0	X	Z	Z	X	Z	Z	XX	Z	Z
0011:	0	C	11	N	N	0	1	X	L	H	X	L	H	XX	L	H
0012:	0	C	11	H	H	0	0	X	N	N	1	H	L	XX	N	N
0013:	0	C	11	H	H	0	0	X	N	N	0	L	H	XX	N	N
0014:	0	C	11	H	H	0	0	X	N	N	0	L	H	XX	N	N
0015:	0	C	11	N	N	0	1	X	L	H	X	L	H	00	L	H
0016:	0	C	11	N	N	0	0	X	N	N	X	N	N	10	H	L
0017:	0	C	11	N	N	0	0	X	N	N	X	N	N	01	L	H
0018:	0	C	11	N	N	0	0	X	N	N	X	N	N	11	H	L
0019:	0	C	11	N	N	0	0	X	N	N	X	N	N	00	H	L

BASIC FLIP-FLOP

Figure 8. JEDEC File

```

CUPL                2.11b Serial# 5-00001-154
Device              g16v8ms Library DLIB-f-23-9
Created             Fri Sep 18 17:57:36 1987
Name                FLIPFLOP
Partno              00001
Revision            01
Date                21/05/87
Designer            ALBERTO
Company             SGS-THOMSON Microelectronics
Assembly            PC AT
Location            PLZN1
*QP20
*QF2194
*G0
*FO
*L0000 11111111111111111111111111111111
*L0032 10111111111111111111111111111111
*L0064 11010111111111111111111111111111
*L0256 11111111111111111111111111111111
*L0288 11111011111111111111111111111111
*L0320 01111101111111111111111111111111
*L0512 11111110111111111111111111111111
*L0544 11111111101101111111111111111011
*L0576 11111111111010111111111111111011
*L0768 11111111111111111111111111111011
*L0800 11111111101110011111111111111111
*L0832 11111111001011111111111111111111
*L1024 11111110111111111111111111111111
*L1056 11111111111111111011111111111011
*L1280 11111111111111111111111111111011
*L1312 11111111101111111101111111111111
*L1536 11111110111111111111111111111111
*L1568 1111111111111111111111111110011011
*L1600 111111111111111111111111111001
*L1792 11111111111111111111111111110111
*L1824 111111111011111111111110111111101
*L1856 111111111011111111111111101011111
*L2048 11111110000000000000000000000000
*L2112 00000000110000001111111111111111
*L2144 11111111111111111111111111111111
*L2176 1111111111111111101
*C5DAA
*18A7
    
```

BASIC FLIP-FLOP

Figure 9. JEDEC File with Test Vectors

```

CUPL                2.11b Serial# 5-00001-154
Device              g16v8ms Library DLIB-f-23-9
Created             Fri Sep 18 17:57:36 1987
Name                FLIPFLOP
Partno              00001
Revision            01
Date                21/05/87
Designer            ALBERTO
Company             SGS-THOMSON Microelectronics
Assembly            PC AT
Location            PLZNI
*QP20
*QF2194
*QV19
*G0
*F0
*L0000 11111111111111111111111111111111
*L0032 10111111111111111111111111111111
*L0064 11010111111111111111111111111111
*L0256 11111111111111111111111111111111
*L0288 11111011111111111111111111111111
*L0320 01111101111111111111111111111111
*L0512 11111110111111111111111111111111
*L0544 11111111101101111111111111111011
*L0576 11111111111010111111111111110111
*L0768 11111111111111111111111111110111
*L0800 11111111101110111111111111111111
*L0832 11111111100101111111111111111111
*L1024 11111111011111111111111111111111
*L1056 11111111111111101111111111111011
*L1280 11111111111111111111111111110111
*L1312 11111111101111111110111111111111
*L1536 11111111011111111111111111111111
*L1568 11111111111111111111111110011011
*L1600 111111111111111111111101111111001
*L1792 11111111111111111111111111110111
*L1824 11111111101111111111101111111101
*L1856 11111111101111111111110101111111
*L2048 11111111000000000000000000000000
*L2112 00000000110000001111111111111111
*L2144 11111111111111111111111111111111
*L2176 111111111111111110:
*C5DAA
*P 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*V0001 X1GXXXXXXNXXNNNNNNHLLN
*V0002 X01XXXXXXNXXNNNNNNLHNN
*V0003 X1XXXXXXNXXNXXNNNNNNLHNN
*V0004 X00XXXXXXNXXNXXNNNNNNHNN
*V0005 X1XXXXXXNXXNXXNNNNNNHNN
*V0006 C111XXXXX0N0LHLLHLLHNNN
*V0007 C1101XXXXX0N0NNNNHLLNNN
*V0008 C1101XXXXX0N0NNNNHLLNNN
*V0009 C1100XXXXX0N0NNNNHLLNNN
*V0010 C110XXXXX0N1ZZZZZZNNN
*V0011 C110XXXXX1N0LHLLHLLHNNN
*V0012 C110X1XX0N0NNHLLHNNHNN
*V0013 C110X0XX0N0NNHLLHNNHNN
*V0014 C110X0XX0N0NNHLLHNNHNN
*V0015 C110XX001N0LHLLHLLHNNN
*V0016 C110XX100N0LHNNNNNNNNN
*V0017 C110XX010N0LHNNNNNNNNN
*V0018 C110XX110N0LHNNNNNNNNN
*V0019 C110XX000N0LHNNNNNNNNN
*AA30
    
```

BASIC FLIP-FLOP

Figure 10. CUPL Expanded Product Terms

```

*****
                                  FLIPFLOP
*****

CUPL          2.11b Serial# 5-00001-154
Device        g16v8ms Library DLIB-f-23-9
Created       Fri Sep 18 17:57:35 1987
Name          FLIPFLOP
Partno        00001
Revision      01
Date          21/05/87
Designer      ALBERTO
Company       SGS Microelettronica
Assembly      PC AT
Location      PLZN1

=====
                          Expanded Product Terms
=====

QDC.d =>
  CLR
  # !PR & !D

QJC.d =>
  CLR
  # QJC & !PR & !J
  # QJT & !PR & K

QSC =>
  !R
  # QSC & S

QTC.d =>
  CLR
  # QTC & !PR & !T
  # QTT & !PR & T

QDT.d =>
  PR
  # !CLR & D

QJT.d =>
  PR
  # !CLR & QJT & !K
  # QJC & !CLR & J

QST =>
  !S
  # QST & R

QTT.d =>
  PR
  # !CLR & QTT & !T
  # !CLR & QTC & T,

QSC.oe =>
  1

QST.oe =>
  1

```

BASIC FLIP-FLOP

Figure 11. Symbol Table

Symbol Table							
Pin Pol	Variable Name	Ext	Pin	Type	Pterms Used	Max Pterms	Min Level
	QDC		14	V	-	-	-
	QDC	d	14	X	2	8	1
	CLK		1	V	-	-	-
	QJC		12	V	-	-	-
	QJC	d	12	X	3	8	1
	CLR		9	V	-	-	-
	QSC		18	V	2	7	1
	QTC		16	V	-	-	-
	QTC	d	16	X	3	8	1
	QDT		15	V	-	-	-
	QDT	d	15	X	2	8	1
	QJT		13	V	-	-	-
	QJT	d	13	X	3	8	1
	OE		11	V	-	-	-
	QST		19	V	2	7	1
	QTT		17	V	-	-	-
	QTT	d	17	X	3	8	1
	PR		4	V	-	-	-
	D		6	V	-	-	-
	J		7	V	-	-	-
	K		8	V	-	-	-
	R		3	V	-	-	-
	S		2	V	-	-	-
	T		5	V	-	-	-
	QSC	oe	18	D	1	1	0
	QST	oe	19	D	1	1	0

LEGEND F : field D : default variable M : extended node
 N : node I : intermediate variable T : function
 V : variable X : extended variable U : undefined

BASIC FLIP-FLOP

Figure 13. Fuse Plot

Syn 2192 x AcG 2133 -

```

Pin #19 2048 Po1 - 2120 Ac1 -
0000 -----x-----
0032 -x-----
0064 --x-x-----
0096 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0128 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0160 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0192 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0224 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #18 2049 Po1 - 2121 Ac1 -
0256 -----x-----
0288 -----x-----
0320 x-----x-----
0352 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0384 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0416 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0448 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0480 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #17 2050 Po1 - 2122 Ac1 x
0512 -----x-----
0544 -----x--x-----x-----
0576 -----x-x-----x-----
0608 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0640 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0672 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0704 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0736 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #16 2051 Po1 - 2123 Ac1 x
0768 -----x-----x-----
0800 -----x--x-----
0832 -----xx-x-----
0864 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0896 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0928 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0960 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
0992 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    
```

```

Pin #15 2052 Po1 - 2124 Ac1 x
1024 -----x-----
1056 -----x-----x-----
1088 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1120 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1152 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1184 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1216 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1248 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #14 2053 Po1 - 2125 Ac1 x
1280 -----x-----x-----
1312 -----x-----x-----
1344 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1376 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1408 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1440 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1472 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1504 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #13 2054 Po1 - 2126 Ac1 x
1536 -----x-----
1568 -----xx--x-----
1600 -----x-----x-----
1632 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1664 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1696 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1728 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1760 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #12 2055 Po1 - 2127 Ac1 x
1792 -----x-----x-----
1824 -----x-----x-----x-----
1856 -----x-----x-x-----
1888 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1920 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1952 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1984 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
2016 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    
```

LEGEND x : fuse not blown
- : fuse blown

BASIC FLIP-FLOP

Figure 14. ABEL Input File and Simulation

```

module    FLIPFLOP
title    ' Basic Flip-flop example
         By Derrick Tuten 13 July 1987
         SGS-THOMSON Microelectronics

" device declaration

" location keyword    device code
  U002    device      'P16V8R';

" pin declaration

" RS latch
  S,R,QST,QSC    pin 2,3,19,18;

" JK flip-flop
  J,K,QJT,QJC    pin 7,8,13,12;

" T flip-flop
  T,QTT,QTC      pin 5,17,16;

" D flip-flop
  D,QDT,QDC      pin 6,15,14;

" Control
  CLK,PR,CLR,OE  pin 1,4,9,11;

equations

" RS flip-flop
  QST =    !S # (R & QST);
  QSC =    !R # (S & QSC);

" JK flip-flop
  QJT :=    PR # (J & QJC & !CLR) # (!K & QJT & !CLR);
  QJC :=    CLR # (!J & QJC & !PR) # (K & QJT & !PR);

" T flip-flop
  QTT :=    PR # (!CLR & !T & QTT) # (!CLR & T & QTC);
  QTC :=    CLR # (!PR & !T & QTC) # (!PR & T & QTT);

" D flip-flop
  QDT :=    PR # (D & !CLR);
  QDC :=    CLR # (!D & !PR);

test_vectors 'Test RS latch'
  ( [S,R] -> [QST,QSC] )
  [0,1] -> [ 1 , 0 ];
  [1,0] -> [ 0 , 1 ];
  [1,1] -> [ 0 , 1 ];
  [1,0] -> [ 0 , 1 ];
  [0,1] -> [ 1 , 0 ];
  [1,1] -> [ 1 , 0 ];

```

BASIC FLIP-FLOP

Figure 14. ABEL Input File and Simulation (Cont'd)

```

test_vectors 'Test JK flip-flop'
  ( [CLK,CLR, PR, J , K ] -> [QJT,QJC] )
  [.C., 0 , 1 ,.X.,.X.] -> [ 1 , 0 ];
  [.C., 1 , 0 ,.X.,.X.] -> [ 0 , 1 ];
  [.C., 0 , 0 , 0 , 1 ] -> [ 0 , 1 ];
  [.C., 0 , 0 , 0 , 0 ] -> [ 0 , 1 ];
  [.C., 0 , 0 , 1 , 1 ] -> [ 1 , 0 ];
  [.C., 0 , 0 , 1 , 0 ] -> [ 1 , 0 ];
  [.C., 0 , 0 , 0 , 0 ] -> [ 1 , 0 ];
  [.C., 0 , 0 , 1 , 1 ] -> [ 0 , 1 ];
  [.C., 0 , 0 , 1 , 0 ] -> [ 1 , 0 ];
  [.C., 0 , 0 , 0 , 1 ] -> [ 0 , 1 ];

test_vectors 'Test T flip-flop'
  ( [CLK,CLR, PR, T ] -> [QTT,QTC] )
  [.C., 0 , 1 ,.X.] -> [ 1 , 0 ];
  [.C., 1 , 0 ,.X.] -> [ 0 , 1 ];
  [.C., 0 , 0 , 0 ] -> [ 0 , 1 ];
  [.C., 0 , 0 , 1 ] -> [ 1 , 0 ];
  [.C., 0 , 0 , 0 ] -> [ 1 , 0 ];
  [.C., 0 , 0 , 1 ] -> [ 0 , 1 ];
  [.C., 0 , 0 , 1 ] -> [ 1 , 0 ];

test_vectors 'Test D flip-flop'
  ( [CLK,CLR, PR, D ] -> [QDT,QDC] )
  [.C., 0 , 1 ,.X.] -> [ 1 , 0 ];
  [.C., 1 , 0 ,.X.] -> [ 0 , 1 ];
  [.C., 0 , 0 , 0 ] -> [ 0 , 1 ];
  [.C., 0 , 0 , 1 ] -> [ 1 , 0 ];
  [.C., 0 , 0 , 1 ] -> [ 1 , 0 ];
  [.C., 0 , 0 , 0 ] -> [ 0 , 1 ];

end FLIPFLOP

```

BASIC FLIP-FLOP

Figure 15. Reduced Equations

ABEL(tm) Version 2.10a - Document Generator
Basic Flip-flop example
By Derrick Tuten 13 July 1987
SGS-THOMSON Microelectronics
Equations for Module FLIPFLOP

Device U002

- Reduced Equations:

QST = (!S # QST & R);

QSC = (!R # QSC & S);

QJT := (PR # !CLR & J & QJC # !CLR & !K & QJT);

QJC := (CLR # !J & !PR & QJC # K & !PR & QJT);

QTT := (PR # !CLR & QTC & T # !CLR & QTT & !T);

QTC := (CLR # !PR & QTC & !T # !PR & QTT & T);

QDT := (PR # !CLR & D);

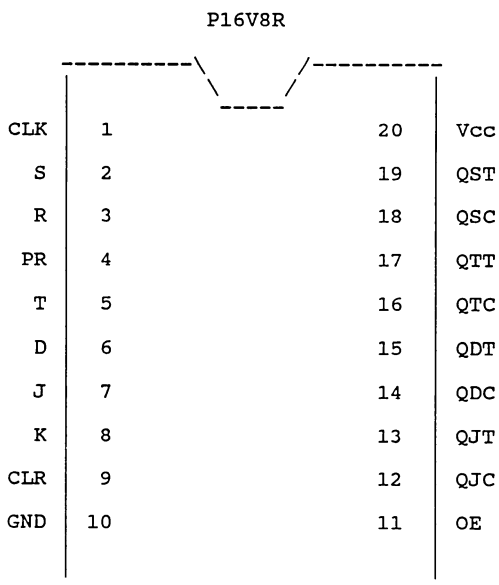
QDC := (CLR # !D & !PR);

BASIC FLIP-FLOP

Figure 16. Chip Diagram

ABEL(tm) Version 2.10a - Document Generator
 Basic Flip-flop example
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 Chip diagram for Module FLIPFLOP

Device U002



BASIC FLIP-FLOP

Figure 17. Chip Usage

ABEL(tm) Version 2.10a - Document Generator
 Basic Flip-flop example
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 for Module FLIPFLOP

Device U002

Device Type: P16V8R

Terms Used: 22 out of 64

Pin #	Name	Terms Used	Max	Term Type	Pin Type
1	CLK	--	--	---	Clock
2	S	--	--	---	Input
3	R	--	--	---	Input
4	PR	--	--	---	Input
5	T	--	--	---	Input
6	D	--	--	---	Input
7	J	--	--	---	Input
8	K	--	--	---	Input
9	CLR	--	--	---	Input
10	GND	--	--	---	GND
11	OE	--	--	---	Enable
12	QJC	3	8	Normal	I/O
13	QJT	3	8	Normal	I/O
14	QDC	2	8	Normal	I/O
15	QDT	2	8	Normal	I/O
16	QTC	3	8	Normal	I/O
17	QTT	3	8	Normal	I/O
18	QSC	2	8	Normal	I/O
19	QST	2	8	Normal	I/O
20	Vcc	--	--	---	VCC

BASIC FLIP-FLOP

Figure 18. ABEL Test Vectors

ABEL(tm) Version 2.10a - Document Generator

Basic Flip-flop example

By Derrick Tuten 13 July 1987

SGS-THOMSON Microelectronics

Test Vectors for Module FLIPFLOP

Device U002

Test RS latch

```

1 [-01- ---- ---- ---- ----] -> [---- ---- ---- ---- -LH-];
2 [-10- ---- ---- ---- ----] -> [---- ---- ---- ---- -HL-];
3 [-11- ---- ---- ---- ----] -> [---- ---- ---- ---- -HL-];
4 [-10- ---- ---- ---- ----] -> [---- ---- ---- ---- -HL-];
5 [-01- ---- ---- ---- ----] -> [---- ---- ---- ---- -LH-];
6 [-11- ---- ---- ---- ----] -> [---- ---- ---- ---- -LH-];

```

Test JK flip-flop

```

7 [C--1 --XX 0--- ---- ----] -> [---- ---- ---L H--- ----];
8 [C--0 --XX 1--- ---- ----] -> [---- ---- ---H L--- ----];
9 [C--0 --01 0--- ---- ----] -> [---- ---- ---H L--- ----];
10 [C--0 --00 0--- ---- ----] -> [---- ---- ---H L--- ----];
11 [C--0 --11 0--- ---- ----] -> [---- ---- ---L H--- ----];
12 [C--0 --10 0--- ---- ----] -> [---- ---- ---L H--- ----];
13 [C--0 --00 0--- ---- ----] -> [---- ---- ---L H--- ----];
14 [C--0 --11 0--- ---- ----] -> [---- ---- ---H L--- ----];
15 [C--0 --10 0--- ---- ----] -> [---- ---- ---L H--- ----];
16 [C--0 --01 0--- ---- ----] -> [---- ---- ---H L--- ----];

```

Test T flip-flop

```

17 [C--1 X--- 0--- ---- ----] -> [---- ---- ---- ---L H---];
18 [C--0 X--- 1--- ---- ----] -> [---- ---- ---- ---H L---];
19 [C--0 0--- 0--- ---- ----] -> [---- ---- ---- ---H L---];
20 [C--0 1--- 0--- ---- ----] -> [---- ---- ---- ---L H---];
21 [C--0 0--- 0--- ---- ----] -> [---- ---- ---- ---L H---];
22 [C--0 1--- 0--- ---- ----] -> [---- ---- ---- ---H L---];
23 [C--0 1--- 0--- ---- ----] -> [---- ---- ---- ---L H---];

```

Test D flip-flop

```

24 [C--1 -X-- 0--- ---- ----] -> [---- ---- ---- -LH-];
25 [C--0 -X-- 1--- ---- ----] -> [---- ---- ---- -HL-];
26 [C--0 -0-- 0--- ---- ----] -> [---- ---- ---- -HL-];
27 [C--0 -1-- 0--- ---- ----] -> [---- ---- ---- -LH-];
28 [C--0 -1-- 0--- ---- ----] -> [---- ---- ---- -LH-];
29 [C--0 -0-- 0--- ---- ----] -> [---- ---- ---- -HL-];

```

end of module FLIPFLOP

BASIC FLIP-FLOP

Figure 19. JEDEC File with Test Vectors

```
ABEL(tm) Version 2.10c FutureNet/Data-IO Corp. JEDEC file for: P16V8R
Created on: 20-Jul-87 01:59 PM
Basic Flip-flop example
By Derrick Tuten 13 July 1987
SGS-THOMSON Microelectronics
QP20* QF2194*
L0000
11111111111111111111111111111111
10111111111111111111111111111111
11010111111111111111111111111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
11111111111111111111111111111111
11111011111111111111111111111111
01111011111111111111111111111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
11111110111111111111111111111111
1111111111010111111111111111011
1111111101101111111111111111011
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
11111111111111111111111111110111
11111111011100111111111111111111
1111111100101111111111111111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
11111110111111111111111111111111
1111111111111101111111111111011
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
11111111111111111111111111110111
11111111011111110111111111111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

BASIC FLIP-FLOP

Figure 19. ABEL Jeduc File with Test Vectors (Cont'd)

```

00000000000000000000000000000000
11111110111111111111111111111111
111111111111111111111011111111001
11111111111111111111111110011011
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
111111111111111111111111111110111
111111110111111111111110111111101
11111111011111111111111010111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000*
L2048
11111111*
L2056
000000000000000000000000000000000000000000000000000000000000000000000000*
L2120
11000000*
L2128
1111111111111111111111111111111111111111111111111111111111111111111111*
L2192
01*
N Test RS latch*
V0001 X01XXXXXXNXXXXXXHLN*
V0002 X10XXXXXXNXXXXXXHLN*
V0003 X11XXXXXXNXXXXXXHLN*
V0004 X10XXXXXXNXXXXXXHLN*
V0005 X01XXXXXXNXXXXXXHLN*
V0006 X11XXXXXXNXXXXXXHLN*
N Test JK flip-flop*
V0007 CXX1XXX0NXHLHXXXXXXN*
V0008 CXX0XXX1NXHLHXXXXXXN*
V0009 CXX0XX010NXHLHXXXXXXN*
V0010 CXX0XX000NXHLHXXXXXXN*
V0011 CXX0XX110NXHLHXXXXXXN*
V0012 CXX0XX100NXHLHXXXXXXN*
V0013 CXX0XX000NXHLHXXXXXXN*
V0014 CXX0XX110NXHLHXXXXXXN*
V0015 CXX0XX100NXHLHXXXXXXN*
V0016 CXX0XX010NXHLHXXXXXXN*
N Test T flip-flop*
V0017 CXX1XXX0NXXXXXXLHXXN*
V0018 CXX0XXX1NXXXXXXLHXXN*
V0019 CXX00XX0NXXXXXXLHXXN*
V0020 CXX01XX0NXXXXXXLHXXN*
V0021 CXX00XX0NXXXXXXLHXXN*
V0022 CXX01XX0NXXXXXXLHXXN*
V0023 CXX01XX0NXXXXXXLHXXN*
N Test D flip-flop*
V0024 CXX1XXX0NXXXXXXLHXXN*
V0025 CXX0XXX1NXXXXXXLHXXN*
V0026 CXX00XX0NXXXXXXLHXXN*
V0027 CXX0X1XX0NXXXXXXLHXXN*
V0028 CXX0X1XX0NXXXXXXLHXXN*
V0029 CXX0X0XX0NXXXXXXLHXXN*
C5DAA*
E61D

```


QUAD 1:1 MULTIPLEXER

A 20V8 Application

Widely used in computer and data communications circuits, multiplexers route one of several input banks to an output, based on the condition of select inputs. This particular version has 4 input banks, each 4-bits wide (figure 1); therefore, two select lines are required to choose 1 of 4 inputs, as shown in the function table of figure 2.

Possible applications for our multiplexer include bus selection in a multibus computer environment, or data manipulation in an arithmetic/logic circuit. With a total of 16 multiplexer inputs and two Select inputs, this design is well suited for the GAL20V8. The Pin Connection chosen for this example is shown in figure 3; actual pin placement of the multiplexer outputs is not critical since the versatility of the GAL20V8 allows the designer to choose that combination of output pins that best suits the board layout. The device was programmed using both CUPL and ABEL.

Figure 2. Function Table

S ₁	S ₀	AOUT	BOUT	COUT	DOUT
0	0	A0	B0	C0	D0
0	1	A1	B1	C1	D1
1	0	A2	B2	C2	D2
1	1	A3	B3	C3	D3

Figure 1. Block Diagram

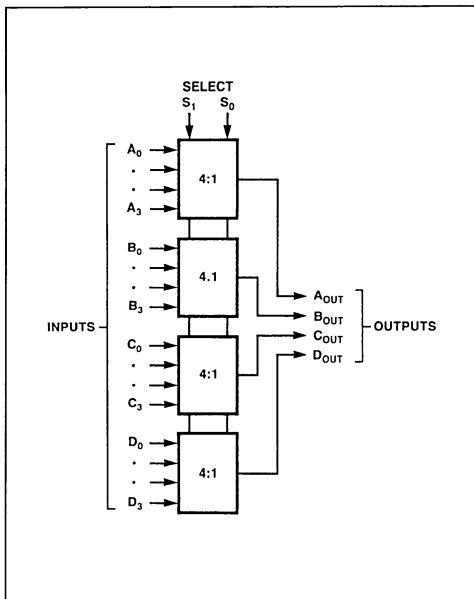
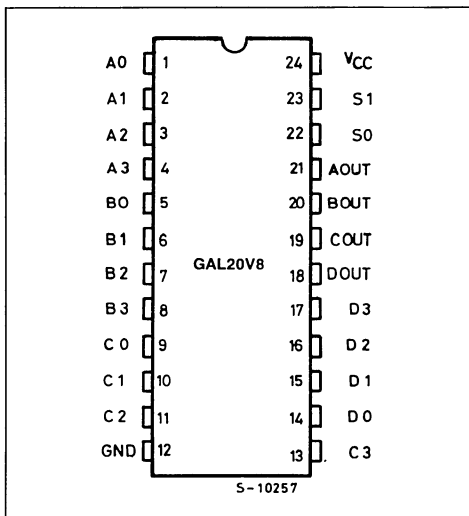


Figure 3. Pin Connection



QUAD 1:1 MULTIPLEXER

Figure 4. CUPL Design Input File

```

Name      QUADMUX;
Partno    00001;
Date      30/06/87;
Revision  01;
Designer  ALBERTO;
Company   SGS-THOMSON Microelectronics
Assembly  PC AT;
Location  PLZN1;
FORMAT    JEDEC;
/*****/
/* QUAD 4:1 MULTIPLEXER */
/*
/*****/
/* Allowable Target Device Types: G20V8 */
/*****/

/** Inputs **/

Pin      [1..4] = [A0..3] ;      /* INPUTS MULTIPLEXER A      */
Pin      [5..8] = [B0..3] ;      /* INPUTS MULTIPLEXER B      */
Pin [9..11,13] = [C0..3] ;      /* INPUTS MULTIPLEXER C      */
Pin      [14..17] = [D0..3] ;    /* INPUTS MULTIPLEXER D      */
Pin      [22,23] = [S0,S1] ;     /* MULTIPLEXER SELECT        */

/** Outputs **/

Pin      [21..18] = [A,B,C,D] ;  /* OUTPUTS MULTIPLEXER      */

/** Declarations and Intermediate Variable Definitions **/

CA0 = !S0 & !S1 ;                /* CONTROL DATA 0          */
CB1 = S0 & !S1 ;                 /* CONTROL DATA 1          */
CC2 = !S0 & S1 ;                 /* CONTROL DATA 2          */
CD3 = S0 & S1 ;                  /* CONTROL DATA 3          */

/** Logic Equations **/

A = (CA0 & A0) # (CB1 & A1) # (CC2 & A2) # (CD3 & A3) ;
B = (CA0 & B0) # (CB1 & B1) # (CC2 & B2) # (CD3 & B3) ;
C = (CA0 & C0) # (CB1 & C1) # (CC2 & C2) # (CD3 & C3) ;
D = (CA0 & D0) # (CB1 & D1) # (CC2 & D2) # (CD3 & D3) ;

```

QUAD 1:1 MULTIPLEXER

Figure 5. CUPL Simulation File

CSIM Version 2.11b Serial# 5-00001-154
 Copyright (C) 1983,1986 Personal CAD Systems, Inc.
 CREATED Fri Sep 18 18:29:31 1987

LISTING FOR SIMULATION FILE: QUADMUX.si

```

1: Name           QUADMUX;
2: Partno        00001;
3: Revision      01;
4: Date          30/06/87;
5: Designer      ALBERTO;
6: Company       SGS-THOMSON Microelectronics
7: Location      PLZN1;
8: Assembly      PC AT;
9: Format         JEDEC;
10: /******
11: /*                                                    */
12: /* QUAD 4:1 MULTIPLEXER                               */
13: /*                                                    */
14: /******
15: /* Target Devices: G20V8                               */
16: /******
17:
18:
19: Order: S0,S1,%2,A0..3,%1,A,%3,B0..3,%1,B,%S,C0..3,%1,C,%S,
        DO..3,%1,D;
20:
21:
    
```

=====
 Simulation Results
 =====

	S0	A0123	A	B0123	B	C0123	C	D0123	D
0001:	00	1000	H	1000	H	1000	H	1000	H
0002:	10	1000	L	1000	L	1000	L	1000	L
0003:	01	1000	L	1000	L	1000	L	1000	L
0004:	11	1000	L	1000	L	1000	L	1000	L
0005:	00	1001	H	1001	H	1001	H	1001	H
0006:	10	1001	L	1001	L	1001	L	1001	L
0007:	01	1001	L	1001	L	1001	L	1001	L
0008:	11	1001	H	1001	H	1001	H	1001	H
0009:	00	0XXX	L	X101	X	X101	X	1XXX	H
0010:	10	0X11	X	X101	H	X101	H	1X00	X

QUAD 1:1 MULTIPLEXER

Figure 6. ABEL Input and Simulation File

```

module    QUAD4T01
title    ' Quad 4 to 1 Multiplexer
          By Derrick Tuten 13 July 1987
          SGS-THOMSON Microelectronics

" device declaration

" location keyword    device code
  U003    device      'P20V8S';

" pin declaration
" inputs
  A0,A1,A2,A3    pin 1,2,3,4;
  B0,B1,B2,B3    pin 5,6,7,8;
  C0,C1,C2,C3    pin 9,10,11,13;
  D0,D1,D2,D3    pin 14,15,16,17;

" outputs
  Aout,Bout      pin 21,20;
  Cout,Dout      pin 19,18;

" control
  S0,S1          pin 22,23;

equations
" A Multiplexer
  Aout =      (!S1 & !S0 & A0) # (!S1 & S0 & A1) #
             (S1 & !S0 & A2) # (S1 & S0 & A3);

" B Multiplexer
  Bout =      (!S1 & !S0 & B0) # (!S1 & S0 & B1) #
             (S1 & !S0 & B2) # (S1 & S0 & B3);

" C Multiplexer
  Cout =      (!S1 & !S0 & C0) # (!S1 & S0 & C1) #
             (S1 & !S0 & C2) # (S1 & S0 & C3);

" D Multiplexer
  Dout =      (!S1 & !S0 & D0) # (!S1 & S0 & D1) #
             (S1 & !S0 & D2) # (S1 & S0 & D3);

test_vectors    'Test A Multiplexer'
  ( [ S1 ,S0 ,A0 ,A1 ,A2 ,A3 ] -> [ Aout ] )
  [ 0 , 0 , 1 , 0 , 0 , 0 ] -> [ 1 ] ;
  [ 0 , 1 , 0 , 1 , 0 , 0 ] -> [ 1 ] ;
  [ 1 , 0 , 0 , 0 , 1 , 0 ] -> [ 1 ] ;
  [ 1 , 1 , 0 , 0 , 0 , 1 ] -> [ 1 ] ;
  [ 0 , 0 , 0 , 0 , 0 , 1 ] -> [ 0 ] ;

test_vectors    'Test B Multiplexer'
  ( [ S1 ,S0 ,B0 ,B1 ,B2 ,B3 ] -> [ Bout ] )
  [ 0 , 0 , 1 , 0 , 0 , 0 ] -> [ 1 ] ;
  [ 0 , 1 , 0 , 1 , 0 , 0 ] -> [ 1 ] ;
  [ 1 , 0 , 0 , 0 , 1 , 0 ] -> [ 1 ] ;
  [ 1 , 1 , 0 , 0 , 0 , 1 ] -> [ 1 ] ;
  [ 0 , 0 , 0 , 0 , 0 , 1 ] -> [ 0 ] ;

```

QUAD 1:1 MULTIPLEXER

Figure 6. ABEL Input and Simulation File (Cont'd)

```

test_vectors 'Test C Multiplexer'
  ( [S1 ,S0 ,C0 ,C1 ,C2 ,C3 ] -> [ Cout] )
  [ 0 , 0 , 1 , 0 , 0 , 0 ] -> [ 1 ] ;
  [ 0 , 1 , 0 , 1 , 0 , 0 ] -> [ 1 ] ;
  [ 1 , 0 , 0 , 0 , 1 , 0 ] -> [ 1 ] ;
  [ 1 , 1 , 0 , 0 , 0 , 1 ] -> [ 1 ] ;
  [ 0 , 0 , 0 , 0 , 0 , 1 ] -> [ 0 ] ;

test_vectors 'Test D Multiplexer'
  ( [S1 ,S0 ,D0 ,D1 ,D2 ,D3 ] -> [ Dout] )
  [ 0 , 0 , 1 , 0 , 0 , 0 ] -> [ 1 ] ;
  [ 0 , 1 , 0 , 1 , 0 , 0 ] -> [ 1 ] ;
  [ 1 , 0 , 0 , 0 , 1 , 0 ] -> [ 1 ] ;
  [ 1 , 1 , 0 , 0 , 0 , 1 ] -> [ 1 ] ;
  [ 0 , 0 , 0 , 0 , 0 , 1 ] -> [ 0 ] ;

end QUAD4TO1

```

Figure 7. ABEL Reduced Equations

ABEL(tm) Version 2.10a - Document Generator
 Quad 4 to 1 Multiplexer
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 Equations for Module QUAD4TO1
 Device U003

- Reduced Equations:

```

Aout = (A0 & !S0 & !S1 # A1 & S0 & !S1 # A2 & !S0 & S1 # A3 & S0 & S1);
Bout = (B0 & !S0 & !S1 # B1 & S0 & !S1 # B2 & !S0 & S1 # B3 & S0 & S1);
Cout = (C0 & !S0 & !S1 # C1 & S0 & !S1 # C2 & !S0 & S1 # C3 & S0 & S1);
Dout = (D0 & !S0 & !S1 # D1 & S0 & !S1 # D2 & !S0 & S1 # D3 & S0 & S1);

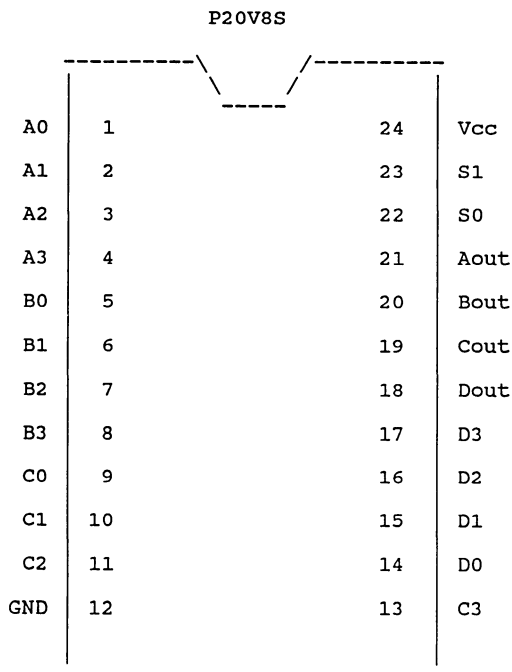
```

QUAD 1:1 MULTIPLEXER

Figure 8. ABEL Chip Diagram

ABEL(™) Version 2.10a - Document Generator
 Quad 4 to 1 Multiplexer
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 Chip diagram for Module QUAD4T01

Device U003



QUAD 1:1 MULTIPLEXER

Figure 9. ABEL Chip Usage

ABEL(tm) Version 2.10a - Document Generator
 Quad 4 to 1 Multiplexer
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 for Module QUAD4T01

Device U003

Device Type: P20V8S

Terms Used: 16 out of 64

Pin #	Name	Terms Used	Max	Term Type	Pin Type
1	A0	--	--	---	Input
2	A1	--	--	---	Input
3	A2	--	--	---	Input
4	A3	--	--	---	Input
5	B0	--	--	---	Input
6	B1	--	--	---	Input
7	B2	--	--	---	Input
8	B3	--	--	---	Input
9	C0	--	--	---	Input
10	C1	--	--	---	Input
11	C2	--	--	---	Input
12	GND	--	--	---	GND
13	C3	--	--	---	Input
14	D0	--	--	---	Input
15	D1	0	8	Normal	I/O
16	D2	0	8	Normal	I/O
17	D3	0	8	Normal	I/O
18	Dout	4	8	Normal	Output
19	Cout	4	8	Normal	Output
20	Bout	4	8	Normal	I/O
21	Aout	4	8	Normal	I/O
22	S0	0	8	Normal	I/O
23	S1	--	--	---	Input
24	Vcc	--	--	---	VCC

QUAD 1:1 MULTIPLEXER

Figure 10. ABEL Test Vectors

ABEL(tm) Version 2.10a - Document Generator
 Quad 4 to 1 Multiplexer
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 Test Vectors for Module QUAD4TO1
 Device U003

Test A Multiplexer

```

1 [1000 ---- ---- ---- ---- -00-] -> [---- ---- ---- ---- ---- H---];
2 [0100 ---- ---- ---- ---- -10-] -> [---- ---- ---- ---- ---- H---];
3 [0010 ---- ---- ---- ---- -01-] -> [---- ---- ---- ---- ---- H---];
4 [0001 ---- ---- ---- ---- -11-] -> [---- ---- ---- ---- ---- H---];
5 [0001 ---- ---- ---- ---- -00-] -> [---- ---- ---- ---- ---- L---];

```

Test B Multiplexer

```

6 [---- 1000 ---- ---- ---- -00-] -> [---- ---- 1000 ---- ---- -H---];
7 [---- 0100 ---- ---- ---- -10-] -> [---- ---- 0100 ---- ---- -H---];
8 [---- 0010 ---- ---- ---- -01-] -> [---- ---- 0010 ---- ---- -H---];
9 [---- 0001 ---- ---- ---- -11-] -> [---- ---- 0001 ---- ---- -H---];
10 [---- 0001 ---- ---- ---- -00-] -> [---- ---- 0001 ---- ---- -L---];

```

Test C Multiplexer

```

11 [---- ---- 100- 0--- ---- -00-] -> [---- ---- ---- ---- ---- -H---];
12 [---- ---- 010- 0--- ---- -10-] -> [---- ---- ---- ---- ---- -H---];
13 [---- ---- 001- 0--- ---- -01-] -> [---- ---- ---- ---- ---- -H---];
14 [---- ---- 000- 1--- ---- -11-] -> [---- ---- ---- ---- ---- -H---];
15 [---- ---- 000- 1--- ---- -00-] -> [---- ---- ---- ---- ---- -L---];

```

Test D Multiplexer

```

16 [---- ---- ---- -100 0--- -00-] -> [---- ---- ---- ---- ---- -H---];
17 [---- ---- ---- -010 0--- -10-] -> [---- ---- ---- ---- ---- -H---];
18 [---- ---- ---- -001 0--- -01-] -> [---- ---- ---- ---- ---- -H---];
19 [---- ---- ---- -000 1--- -11-] -> [---- ---- ---- ---- ---- -H---];
20 [---- ---- ---- -000 1--- -00-] -> [---- ---- ---- ---- ---- -L---];

```

end of module QUAD4TO1

QUAD 1:1 MULTIPLEXER

Figure 11. JEDEC File with Test Vectors (Cont'd)

```
V0011 XXXXXXXX100N0XXXXNHXX00N*
V0012 XXXXXXXX010N0XXXXNHXX10N*
V0013 XXXXXXXX001N0XXXXNHXX01N*
V0014 XXXXXXXX000N1XXXXNHXX11N*
V0015 XXXXXXXX000N1XXXXNLXX00N*
N Test D Multiplexer*
V0016 XXXXXXXXXXXXNX1000HNXX00N*
V0017 XXXXXXXXXXXXNX0100HNXX10N*
V0018 XXXXXXXXXXXXNX0010HNXX01N*
V0019 XXXXXXXXXXXXNX0001HNXX11N*
V0020 XXXXXXXXXXXXNX0001LNXX00N*
C5127*
230E
```

SEVEN SEGMENT DISPLAY DRIVER

Using abel

Here a GAL16V8 is used to drive a seven segment display. Included in the GAL is a decimal up-down counter with asynchronous carry-out and reset functions. The clock and carry-out pins make the counter fully cascadable to form large counters. The ABEL design input files and simulation file are

shown in figure 1.

Note that the counter requires seven registers and the asynchronous output, taking full advantage of the GAL architecture.

The JEDEC file with fuse map and test vectors is shown in figure 6.

Figure 1. ABEL Input and Simulation File

```

module      segcnt7
title '    Seven Segment Display Counter
        By Derrick Tuten 13 July 1987
        SGS-THOMSON Microelectronics
"
"      a
"      ---          Seven-Segment Counter; counts from 0 - 9
"      f| g |b      and loops with a carry out at 9 for cascading
"      ---          displays.
"      e| d |c      <- Segement identification.
"      ---
"      ---          ---          ---          ---          ---          ---
"      | | | | | | | | | | | | | | | | | | | | | | | | | | | |
"      | | | | | | | | | | | | | | | | | | | | | | | | | | | |
"      | | | | | | | | | | | | | | | | | | | | | | | | | | | |
"      ---          ---          ---          ---          ---          ---
"
" device declaration
" location keyword device code
"         U005 device 'P16V8R';
"
" pin declaration
" inputs
"   clk,clr,dir,OE   pin 1,2,3,11;
" outputs
"   a,b,c,d,e,f,g   pin 19,18,17,16,15,14,13;
"   carry_out       pin 12;
"
" C,X,Z = .C., .X., .Z.;
"
" Counter States
ZERO = ^b1111110;
ONE  = ^b0110000;
TWO  = ^b1101101;
THREE = ^b1111001;
FOUR  = ^b0110011;
FIVE  = ^b1011011;
SIX   = ^b1011111;
SEVEN = ^b1110000;
EIGHT = ^b1111111;
NINE  = ^b1111011;
"
" Counter Modes
mode = [clr,dir];
up   = [ 1 , 1 ];
down = [ 1 , 0 ];
clear = [ 0 , X ];

```

SEVEN SEGMENT DISPLAY DRIVER

Figure 1. ABEL Input and Simulation File (Cont'd)

```

equations
    ![a,b,c,d,e,f,g] := (mode == clear) & ![1,1,1,1,1,0];
    carry_out = (mode != clear) & a & b & c & d & !e & f & g;

test_vectors ( [clk, mode, OE] -> [a,b,c,d,e,f,g])
    [ C ,clear, 0 ] -> ZERO;
    [ C , up , 0 ] -> ONE;
    [ C , up , 0 ] -> TWO;
    [ C , up , 0 ] -> THREE;
    [ C , up , 0 ] -> FOUR;
    [ C , up , 0 ] -> FIVE;
    [ C , up , 0 ] -> SIX;
    [ C , up , 0 ] -> SEVEN;
    [ C , up , 0 ] -> EIGHT;
    [ C , up , 0 ] -> NINE;
    [ C , up , 0 ] -> ZERO;
    [ C ,down, 0 ] -> NINE;
    [ C ,down, 0 ] -> EIGHT;
    [ C ,down, 1 ] -> Z;
    [ C ,down, 0 ] -> SIX;
    [ C ,down, 0 ] -> FIVE;
    [ C ,down, 0 ] -> FOUR;
    [ C ,down, 0 ] -> THREE;
    [ C ,down, 0 ] -> TWO;
    [ C ,down, 0 ] -> ONE;

state_diagram [a,b,c,d,e,f,g]
    State ZERO : case (mode == up) : ONE;
                  (mode == down) : NINE;
                endcase;
    State ONE : case (mode == up) : TWO;
                  (mode == down) : ZERO;
                endcase;
    State TWO : case (mode == up) : THREE;
                  (mode == down) : ONE;
                endcase;
    State THREE : case (mode == up) : FOUR;
                   (mode == down) : TWO;
                 endcase;
    State FOUR : case (mode == up) : FIVE;
                   (mode == down) : THREE;
                 endcase;
    State FIVE : case (mode == up) : SIX;
                   (mode == down) : FOUR;
                 endcase;
    State SIX : case (mode == up) : SEVEN;
                  (mode == down) : FIVE;
                endcase;
    State SEVEN : case (mode == up) : EIGHT;
                   (mode == down) : SIX;
                 endcase;
    State EIGHT : case (mode == up) : NINE;
                   (mode == down) : SEVEN;
                 endcase;
    State NINE : case (mode == up) : ZERO;
                  (mode == down) : EIGHT;
                endcase;

end segcnt7

```

SEVEN SEGMENT DISPLAY DRIVER

Figure 2. ABEL Reduced Equations

ABEL(tm) Version 2.10a - Document Generator
 Seven Segment Display Counter
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 Equations for Module segcnt7

Device U005

- Reduced Equations:

```

a := !(a & !b & c & clr & d & !dir & !e & f & g
      # a & b & !c & clr & d & !dir & e & !f & g
      # a & b & c & clr & d & dir & !e & !f & g
      # a & b & c & clr & d & dir & e & f & !g);

b := !(!a & b & c & clr & !d & dir & !e & f & g
      # a & !b & c & clr & d & !dir & e & f & g
      # a & !b & c & clr & d & dir & !e & f & g
      # a & b & c & clr & !d & !dir & !e & !f & !g);

c := !(!a & b & c & clr & !d & dir & !e & !f & !g
      # a & b & c & clr & d & !dir & !e & !f & g);

d := !(a & !b & c & clr & d & !dir & !e & f & g
      # a & b & !c & clr & d & !dir & e & !f & g
      # a & b & c & clr & d & dir & !e & !f & g
      # a & b & c & clr & d & dir & e & f & !g
      # a & !b & c & clr & d & dir & e & f & g
      # a & b & c & clr & d & !dir & e & f & g);

e := !(a & !b & c & clr & d & !dir & !e & f & g
      # a & b & c & clr & d & dir & !e & !f & g
      # !a & b & c & clr & !d & dir & !e & f & g
      # !a & b & c & clr & !d & !dir & !e & f & g
      # a & b & !c & clr & d & e & !f & g
      # a & b & c & clr & d & e & f
      # a & c & clr & d & e & f & g);

f := !(a & b & c & clr & d & dir & e & f & !g
      # !a & b & c & clr & !d & dir & !e & !f & !g
      # a & b & c & clr & d & !dir & !e & !f & g
      # a & !b & c & clr & d & dir & e & f & g
      # a & b & c & clr & d & !dir & e & f & g
      # !a & b & c & clr & !d & !dir & !e & f & g
      # a & b & !c & clr & d & e & !f & g);

g := !(a & b & !c & clr & d & !dir & e & !f & g
      # a & b & c & clr & d & dir & e & f & !g
      # a & !b & c & clr & d & dir & e & f & g
      # a & b & c & clr & d & !dir & e & f & g
  
```

SEVEN SEGMENT DISPLAY DRIVER

Figure 2. ABEL Reduced Equations (Cont'd)

ABEL(tm) Version 2.10a - Document Generator
 Seven Segment Display Counter
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 Equations for Module segcnt7

Device U005

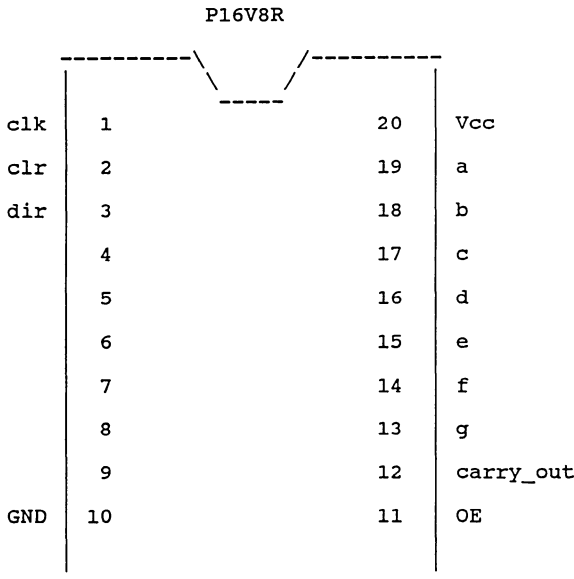
```
# !clr
# !a & b & c & !d & !dir & !e & !f & !g
# a & b & c & d & dir & !e & f & g);
```

```
carry_out = (a & b & c & clr & d & !e & f & g);
```

Figure 3. Chip Diagram

ABEL(tm) Version 2.10a - Document Generator
 Seven Segment Display Counter
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 Chip diagram for Module segcnt7

Device U005



SEVEN SEGMENT DISPLAY DRIVER

Figure 4. Chip Usage

ABEL(tm) Version 2.10a - Document Generator
 Seven Segment Display Counter
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 for Module segcnt7

Device U005

Device Type: P16V8R

Terms Used: 39 out of 64

Pin #	Name	Terms		Term Type	Pin Type
		Used	Max		
1	clk	--	--	---	Clock
2	clr	--	--	---	Input
3	dir	--	--	---	Input
4		--	--	---	Input
5		--	--	---	Input
6		--	--	---	Input
7		--	--	---	Input
8		--	--	---	Input
9		--	--	---	Input
10	GND	--	--	---	GND
11	OE	--	--	---	Enable
12	carry_out	1	8	Normal	I/O
13	g	7	8	Normal	I/O
14	f	7	8	Normal	I/O
15	e	7	8	Normal	I/O
16	d	6	8	Normal	I/O
17	c	2	8	Normal	I/O
18	b	4	8	Normal	I/O
19	a	4	8	Normal	I/O
20	Vcc	--	--	---	VCC

SEVEN SEGMENT DISPLAY DRIVER

Figure 5. ABEL Test Vectors

ABEL(tm) Version 2.10a - Document Generator
Seven Segment Display Counter
By Derrick Tuten 13 July 1987
SGS-THOMSON Microelectronics
Test Vectors for Module segcnt7

Device U005

```
1 [COX- ---- -0- ---- ----] -> [---- ---- ---- LHHH HHH-];
2 [C11- ---- -0- ---- ----] -> [---- ---- ---- LLLL HHL-];
3 [C11- ---- -0- ---- ----] -> [---- ---- ---- HLHH LHH-];
4 [C11- ---- -0- ---- ----] -> [---- ---- ---- HLLH HHH-];
5 [C11- ---- -0- ---- ----] -> [---- ---- ---- HHLL HHL-];
6 [C11- ---- -0- ---- ----] -> [---- ---- ---- HHLH HLH-];
7 [C11- ---- -0- ---- ----] -> [---- ---- ---- HHHH HLH-];
8 [C11- ---- -0- ---- ----] -> [---- ---- ---- LLLL HHH-];
9 [C11- ---- -0- ---- ----] -> [---- ---- ---- HHHH HHH-];
10 [C11- ---- -0- ---- ----] -> [---- ---- ---- HHLH HHH-];
11 [C11- ---- -0- ---- ----] -> [---- ---- ---- LHHH HHH-];
12 [C10- ---- -0- ---- ----] -> [---- ---- ---- HHLH HHH-];
13 [C10- ---- -0- ---- ----] -> [---- ---- ---- HHHH HHH-];
14 [C10- ---- -1- ---- ----] -> [---- ---- ---- ZZZZ ZZZ-];
15 [C10- ---- -0- ---- ----] -> [---- ---- ---- HHHH HLH-];
16 [C10- ---- -0- ---- ----] -> [---- ---- ---- HHLH HLH-];
17 [C10- ---- -0- ---- ----] -> [---- ---- ---- HLLH HHL-];
18 [C10- ---- -0- ---- ----] -> [---- ---- ---- HLLH HHH-];
19 [C10- ---- -0- ---- ----] -> [---- ---- ---- HLHH LHH-];
20 [C10- ---- -0- ---- ----] -> [---- ---- ---- LLLL HHL-];
```

end of module segcnt7

SEVEN SEGMENT DISPLAY DRIVER

Figure 6. ABEL JEDEC File with Test Vectors and Fuse Map

ABEL(tm) Version 2.10c FutureNet/Data-IO Corp. JEDEC file for: P16V8R
 Created on: 20-Jul-87 01:25 PM
 Seven Segment Display Counter
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics

```
QP20* QF2194*
L0000
01011010110111011110110111011111
01011001111011011101111011011111
01010101110111011110111011011111
01010101110111011101110111101111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
01100101110111011101110111011111
01011010110111011101110111011111
01010110110111011101110111011111
01011001110111011101110111011111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
01100101110111011101110111011111
01011001110111011101110111011111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
01011010110111011101110111011111
0101100111101101110111101110111111
01010101110111011101110111011111
0101010111011101110111011110111111
010101101101110111011101110111111
01011001110111011101110111011111
00000000000000000000000000000000
00000000000000000000000000000000
01011010110111011101110111011111
010101011101110111011101110111111
011001011101110111011101110111111
011010011101110111011101110111111
0101110111101101110111101110111111
0101110111011101110111011111111111
0101111111101110111011101110111111
00000000000000000000000000000000
0101010111011101110111011101111111
0110010111011110111011101110111111
010110011101110111011101110111111
010101101101110111011101110111111
010110011101110111011101110111111
011010011101110111011101110111111
0101110111101101110111101110111111
```


7-BIT COUNTER

In this example a GAL20V8 implements a seven-bit counter with asynchronous carry-out and load functions. As illustrated in the block diagram (figure 1) and Pin Connections diagram (figure 2), the carry-in and carry-out pins make the counter fully cascadable to form larger counters. The CUPL de-

sign input file is shown in figure 3, and simulation files in figure 4, while the ABEL design input file and simulation file are shown in figure 5. Note that the counter requires seven register and one asynchronous output, taking full advantage of the generic architecture of the GAL20V8.

Figure 1. Block Diagram

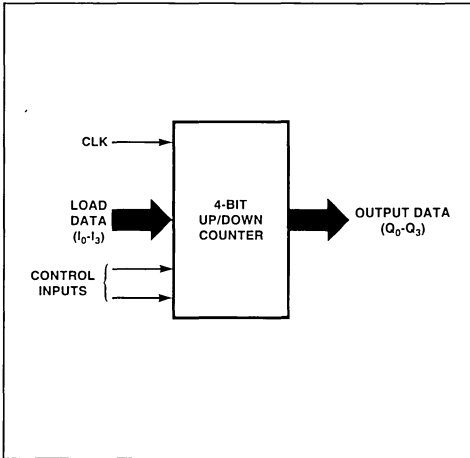
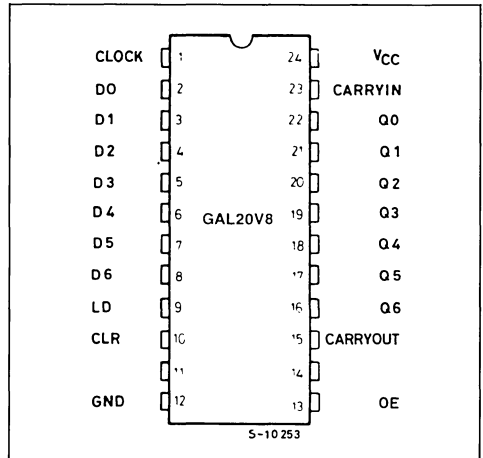


Figure 2. Pin Connections



7-BIT COUNTER

Figure 3. CUPL Design Input File

```

Name          COUNTER;
Partno        00001;
Date          01/07/87;
Revision      01;
Designer      ALBERTO;
Company       SGS-THOMSON Microelectronics
Assembly     PC AT;
Location      PLZN1;
Device        G20V8;
Format        JEDEC;
/*****
/*
/* 7-BIT COUNTER WITH LOAD AND CARRY : INPUT FILE          */
/*
*****/
/* Allowable Target Device Types : G20V8                  */
*****/

/** Inputs **/

Pin    [2..8] = [D0..6] ;                /* INPUT D0..6      */
Pin    [1,13,10] = [CLK,OE,CLR] ;        /* CONTROL INPUT    */
Pin    [9,23] = [LD,CIN] ;              /* LOAD AND CARRY-IN */

/** Outputs **/

Pin    [22..16] = [Q0..6] ;              /* OUTPUT Q0..6     */
Pin    15 = COUT ;                       /* CARRY OUTPUT      */

/** Declarations and Intermediate Variable Definitions **/

QTOT = Q6 & Q5 & Q4 & Q3 & Q2 & Q1 & Q0 ;

/** Logic Equations **/
Q0.D = (LD & D0
        # !LD & !Q0 & CIN) & !CLR ;
Q1.D = (LD & D1
        # !LD & !Q1 & Q0 & CIN
        # !LD & Q1 & ! Q0) & ! CLR ;
Q2.D = (LD & D2
        # !LD & !Q2 & Q1 & Q0 & CIN
        # !LD & Q2 & (!Q1 # !Q0)) & !CLR ;
Q3.D = (LD & D3
        # !LD & ! Q3 & Q2 & Q1 & Q0 & CIN
        # !LD & Q3 & (!Q2 # !Q1 # !Q0)) & !CLR ;
Q4.D = (LD & D4
        # !LD & !Q4 & Q3 & Q2 & Q1 & Q0 & CIN
        # !LD & Q4 & (!Q3 # !Q2 # !Q1 # !Q0)) & !CLR ;
Q5.D = (LD & D5
        # !LD & !Q5 & Q4 & Q3 & Q2 & Q1 & Q0 & CIN
        # !LD & Q5 & (!Q4 # !Q3 # !Q2 # !Q1 # !Q0)) & !CLR ;
Q6.D = (LD & D6
        # !LD & !Q6 & Q5 & Q4 & Q3 & Q2 & Q1 & Q0 & CIN
        # !LD & Q6 & (!Q5 # !Q4 # !Q3 # !Q2 # !Q1 # !Q0)) & !CLR ;

COUT = !LD & QTOT & CIN ;

```

7-BIT COUNTER

Figure 4. CUPL Simulation File

CSIM Version 2.11b Serial# 5-00001-154
 Copyright (C) 1983,1986 Personal CAD Systems, Inc.
 CREATED Fri Sep 18 18:32:07 1987

LISTING FOR SIMULATION FILE: COUNTER.si

```

1: Name           COUNTER;
2: Partno        00001;
3: Revision      01;
4: Date          01/07/87;
5: Designer      ALBERTO;
6: Company       SGS-THOMSON Microelectronics
7: Location      PLZN1;
8: Assembly      PC AT;
9: Format         JEDEC;
10: /******//
11: /*           */
12: /* 7-BIT COUNTER WITH LOAD AND CARRY : SIMULATION FILE */
13: /*           */
14: /******//
15: /* Target Devices: G20V8 */
16: /******//
17:
18:
19: Order: OE,%3,CLK,%3,CLR,%3,LD,%2,D0..6,%2,Q0..6,%2,COU,%3,CIN;
20:
21:

```

```

=====
Simulation Results
=====

```

```

=====
OE CLK CLR LD D0123456 Q0123456 COU CIN
=====

```

```

0001: 0 C 1 0 XXXXXXXX LLLLLLLL L 1
0002: 0 C 0 1 0001111 LLLHHHH L 1
0003: 0 C 0 0 XXXXXXXX HLLHHHH L 1
0004: 0 C 0 0 XXXXXXXX LHLHHHH L 1
0005: 0 C 0 0 XXXXXXXX HHLHHHH L 1
0006: 0 C 0 0 XXXXXXXX LLHHHHH L 1
0007: 0 C 0 0 XXXXXXXX HLHHHHH L 1
0008: 0 C 0 0 XXXXXXXX LHHHHHH L 1
0009: 0 C 0 0 XXXXXXXX HHHHHHH H 1
0010: 0 C 0 0 XXXXXXXX LLLLLLLL L 1
0011: 0 C 0 0 XXXXXXXX HLLLLLLL L 1
0012: 0 C 0 0 XXXXXXXX LHLLLLL L 1
0013: 1 C 0 0 0000000 ZZZZZZZ L 1
0014: 0 C 0 0 0000000 LLHLLLL L 1
0015: 0 C 0 0 0000000 HLHLLLL L 1
0016: 0 C 0 1 1010101 HLHLHLH L 1
0017: 0 C 0 0 XXXXXXXX LLHLHLH L 0
0018: 0 C 0 0 XXXXXXXX LLHLHLH L 0

```

7-BIT COUNTER

Figure 5. ABEL Input and Simulation Files

```

module BITCNTR7
title ' 7 Bit Counter with Asynchronous
      Carry-out and Load Functions.
      By Derrick Tuten 13 July 1987
      SGS-THOMSON Microelectronics
" device declaration

" location keyword   device code
  U004   device     'P20V8R';

" pin declaration
" inputs
  D0,D1,D2,D3   pin 2,3,4,5;
  D4,D5,D6     pin 6,7,8;

" outputs
  Q0,Q1,Q2,Q3   pin 22,21,20,19;
  Q4,Q5,Q6     pin 18,17,16;

" control
  CLK,LD,CLR    pin 1,9,10;
  CARRYIN      pin 23;
  CARRYOUT     pin 15;
  OE           pin 13;

" declarations
  C,H,L,X,Z    = .C.,1,0,.X.,.Z.;

equations
Q0 := (LD & D0 #                               " LOAD D0
      !LD & !Q0 & CARRYIN) & !CLR;          " TOGGLE

Q1 := (LD & D1 #                               " LOAD D1
      !LD & !Q1 & Q0 & CARRYIN #           " TOGGLE
      !LD & Q1 & !Q0) & !CLR;              " HOLD

Q2 := (LD & D2 #                               " LOAD D2
      !LD & !Q2 & Q1 & Q0 & CARRYIN #       " TOGGLE
      !LD & Q2 & !Q1 #                       " HOLD
      !LD & Q2 & !Q0) & !CLR;              " HOLD

Q3 := (LD & D3 #                               " LOAD D3
      !LD & !Q3 & Q2 & Q1 & Q0 & CARRYIN #   " TOGGLE
      !LD & Q3 & !Q2 #                       " HOLD
      !LD & Q3 & !Q1 #                       " HOLD
      !LD & Q3 & !Q0) & !CLR;              " HOLD

Q4 := (LD & D4 #                               " LOAD D4
      !LD & !Q4 & Q3 & Q2 & Q1 & Q0 & CARRYIN # " TOGGLE
      !LD & Q4 & !Q3 #                       " HOLD
      !LD & Q4 & !Q2 #                       " HOLD
      !LD & Q4 & !Q1 #                       " HOLD
      !LD & Q4 & !Q0) & !CLR;              " HOLD

```

7-BIT COUNTER

Figure 5. ABEL Input and Simulation Files (Cont'd)

```

Q5 := (LD & D5 # " LOAD D5
      !LD & !Q5 & Q4 & Q3 & Q2 & Q1 & Q0 & CARRYIN # " TOGGLE
      !LD & Q5 & !Q4 # " HOLD
      !LD & Q5 & !Q3 # " HOLD
      !LD & Q5 & !Q2 # " HOLD
      !LD & Q5 & !Q1 # " HOLD
      !LD & Q5 & !Q0) & !CLR; " HOLD

Q6 := (LD & D6 # " LOAD D6
      !LD & !Q6 & Q5 & Q4 & Q3 & Q2 & Q1 & Q0 & CARRYIN # " TOGGLE
      !LD & Q6 & !Q5 # " HOLD
      !LD & Q6 & !Q4 # " HOLD
      !LD & Q6 & !Q3 # " HOLD
      !LD & Q6 & !Q2 # " HOLD
      !LD & Q6 & !Q1 # " HOLD
      !LD & Q6 & !Q0) & !CLR; " HOLD

CARRYOUT = !LD & Q6 & Q5 & Q4 & Q3 & Q2 & Q1 & Q0 & CARRYIN; " CARRYOUT

test_vectors ' Test Counter '

([CLK,OE,CLR,LD,CARRYIN,D6,D5,D4,D3,D2,D1,D0] ->
 [Q6,Q5,Q4,Q3,Q2,Q1,Q0,CARRYOUT])

[0, 1, X, X, X, X,X,X,X,X,X,X] -> [Z,Z,Z,Z,Z,Z,Z, X];
[C, 0, 1, X, X, X,X,X,X,X,X,X] -> [L,L,L,L,L,L,L, L];
[C, 0, 0, 1, X, 1,1,1,1,1,1,1] -> [H,H,H,H,H,H,H, L];
[C, 0, 0, 1, X, 0,0,0,0,0,0,0] -> [L,L,L,L,L,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,L,L,L,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,L,L,L,H,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,L,L,L,H,H, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,L,L,L,H,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,L,L,H,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,L,L,H,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,L,L,H,H,H, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,L,H,L,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,L,H,L,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,L,H,H,H,H, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,H,L,L,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,H,L,L,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,H,L,L,H,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,H,L,H,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,H,L,H,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,H,L,H,H,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,H,H,L,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,H,H,L,H,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,H,H,L,H,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X,X] -> [L,L,H,H,L,H,L, L];

```


7-BIT COUNTER

Figure 5. ABEL Input and Simulation Files (Cont'd)

```
[C, 0, 0, 0, 1, X,X,X,X,X,X,X] -> [L,L,H,H,H,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X] -> [L,L,H,H,H,L,H, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X] -> [L,L,H,H,H,H,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X] -> [L,L,H,H,H,H,H, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X] -> [L,H,L,L,L,L,L, L];
[C, 0, 0, 1, X, 0,1,1,1,1,1,1] -> [L,H,H,H,H,H,H, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X] -> [H,L,L,L,L,L,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X] -> [H,L,L,L,L,L,H, L];
[C, 0, 0, 1, X, 1,1,1,1,1,1,0] -> [H,H,H,H,H,H,L, L];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X] -> [H,H,H,H,H,H,H, H];
[C, 0, 0, 0, 1, X,X,X,X,X,X,X] -> [L,L,L,L,L,L,L, L];
```

end BITCNTR7

7-BIT COUNTER

Figure 6. ABEL Reduced Equations

ABEL(tm) Version 2.10a - Document Generator
 7 Bit Counter with Asynchronous
 Carry-out and Load Functions.
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 Equations for Module BITCNR7

Device U004

- Reduced Equations:

```

Q0 := (!CLR & D0 & LD # CARRYIN & !CLR & !LD & !Q0);

Q1 := (!CLR & D1 & LD
      # !CLR & !LD & !Q0 & Q1
      # CARRYIN & !CLR & !LD & Q0 & !Q1);

Q2 := (!CLR & D2 & LD
      # !CLR & !LD & !Q0 & Q2
      # !CLR & !LD & !Q1 & Q2
      # CARRYIN & !CLR & !LD & Q0 & Q1 & !Q2);

Q3 := (!CLR & D3 & LD
      # !CLR & !LD & !Q0 & Q3
      # !CLR & !LD & !Q1 & Q3
      # !CLR & !LD & !Q2 & Q3
      # CARRYIN & !CLR & !LD & Q0 & Q1 & Q2 & !Q3);

Q4 := (!CLR & D4 & LD
      # !CLR & !LD & !Q0 & Q4
      # !CLR & !LD & !Q1 & Q4
      # !CLR & !LD & !Q2 & Q4
      # !CLR & !LD & !Q3 & Q4
      # CARRYIN & !CLR & !LD & Q0 & Q1 & Q2 & Q3 & !Q4);

Q5 := (!CLR & D5 & LD
      # !CLR & !LD & !Q0 & Q5
      # !CLR & !LD & !Q1 & Q5
      # !CLR & !LD & !Q2 & Q5
      # !CLR & !LD & !Q3 & Q5
      # !CLR & !LD & !Q4 & Q5
      # CARRYIN & !CLR & !LD & Q0 & Q1 & Q2 & Q3 & Q4 & !Q5);

Q6 := (!CLR & D6 & LD
      # !CLR & !LD & !Q0 & Q6
      # !CLR & !LD & !Q1 & Q6
      # !CLR & !LD & !Q2 & Q6
      # !CLR & !LD & !Q3 & Q6
      # !CLR & !LD & !Q4 & Q6
      # !CLR & !LD & !Q5 & Q6
      # CARRYIN & !CLR & !LD & Q0 & Q1 & Q2 & Q3 & Q4 & Q5 & !Q6);

```


7-BIT COUNTER

Figure 8. ABEL Chip Usage

ABEL(tm) Version 2.10a - Document Generator
 7 Bit Counter with Asynchronous
 Carry-out and Load Functions.
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 for Module BITCNTR7

Device U004

Device Type: P20V8R

Terms Used: 37 out of 64

Pin #	Name	Terms			Pin Type
		Used	Max	Term Type	
1	CLK	--	--	---	Clock
2	D0	--	--	---	Input
3	D1	--	--	---	Input
4	D2	--	--	---	Input
5	D3	--	--	---	Input
6	D4	--	--	---	Input
7	D5	--	--	---	Input
8	D6	--	--	---	Input
9	LD	--	--	---	Input
10	CLR	--	--	---	Input
11		--	--	---	Input
12	GND	--	--	---	GND
13	OE	--	--	---	Enable
14		--	--	---	Input
15	CARRYOUT	1	8	Normal	I/O
16	Q6	8	8	Normal	I/O
17	Q5	7	8	Normal	I/O
18	Q4	6	8	Normal	I/O
19	Q3	5	8	Normal	I/O
20	Q2	4	8	Normal	I/O
21	Q1	3	8	Normal	I/O
22	Q0	2	8	Normal	I/O
23	CARRYIN	--	--	---	Input
24	Vcc	--	--	---	VCC

7-BIT COUNTER

Figure 9. Test Vectors

ABEL(tm) Version 2.10a - Document Generator
 7 Bit Counter with Asynchronous
 Carry-out and Load Functions.
 By Derrick Tuten 13 July 1987
 SGS-THOMSON Microelectronics
 Test Vectors for Module BITCNTR7

Device U004

Test Counter

```

1 [0XXX XXXX XX-- 1--- ---- --X-] -> [---- ---- ---- --XZ ZZZZ ZZ--];
2 [CXXX XXXX X1-- 0--- ---- --X-] -> [---- ---- ---- --LL LLLL LL--];
3 [C111 1111 10-- 0--- ---- --X-] -> [---- ---- ---- --LH HHHH HH--];
4 [C000 0000 10-- 0--- ---- --X-] -> [---- ---- ---- --LL LLLL LL--];
5 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLLL LH--];
6 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLLL HL--];
7 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLLL HH--];
8 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLLH LL--];
9 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLLH LH--];
10 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLLH HL--];
11 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLLH HH--];
12 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLHL LL--];
13 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLHL LH--];
14 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLHL HL--];
15 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLHL HH--];
16 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLHH LL--];
17 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLHH LH--];
18 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLHH HL--];
19 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLHH HH--];
20 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHLH LL--];
21 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHLH LH--];
22 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHLH HL--];
23 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHLH HH--];
24 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHLH LL--];
25 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHLH LH--];
26 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHLH HL--];
27 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHLH HH--];
28 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHLH LL--];
29 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHLH LH--];
30 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHLH HL--];
31 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHLH HH--];
32 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHHH LL--];
33 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHHH LH--];
34 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHHH HL--];
35 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LHHH HH--];
36 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLLL LL--];
37 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLLL LH--];
38 [C111 1110 10-- 0--- ---- --X-] -> [---- ---- ---- --LL HHHH HH--];
39 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LH LLLL LL--];
40 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LH LLLL LH--];
41 [C011 1111 10-- 0--- ---- --X-] -> [---- ---- ---- --LH HHHH HL--];
42 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --HH HHHH HH--];
43 [CXXX XXXX 00-- 0--- ---- --1-] -> [---- ---- ---- --LL LLLL LL--];
    
```

end of module BITCNTR7

7-BIT COUNTER

Figure 10. ABEL JEDEC File with Test Vectors (Cont'ed)

```

110111011101110111011101110111011101110111111
00000000000000000000000000000000000000000000
11111111111111111111111111011101110111011111
11111101111111111111111111111111001101111111
1111111111011111111111111111001101111111
111111111111101111111111111111001101111111
11111111111111111111110111111111110011011111
11111111111111111111110111111111110011011111
1111111111111111111111111101001101111111
11011011101110111011101110111011010101111111
11111111111111111111111111111111111111111111
11011011101110111011101110111011001111111111
00000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000*
L2560
11111111*
L2568
00000000000000000000000000000000000000000000*
L2632
00000001*
L2640
11111111111111111111111111111111111111111111*
L2704
01*
N Test Counter *
V0001 0XXXXXXXXXN1XXZZZZZXN*
V0002 CXXXXXXXX1XNOXL L L L L L L L L XN*
V0003 C111111110XNOXL H H H H H H H XN*
V0004 C000000010XNOXL L L L L L L L L XN*
V0005 CXXXXXXXX00XNOXL L L L L L L L H1N*
V0006 CXXXXXXXX00XNOXL L L L L L L L H1N*
V0007 CXXXXXXXX00XNOXL L L L L L L H H1N*
V0008 CXXXXXXXX00XNOXL L L L L L L H L1N*
V0009 CXXXXXXXX00XNOXL L L L L L L H L1N*
V0010 CXXXXXXXX00XNOXL L L L L L H H L1N*
V0011 CXXXXXXXX00XNOXL L L L L L H H H1N*
V0012 CXXXXXXXX00XNOXL L L L L L H L L1N*
V0013 CXXXXXXXX00XNOXL L L L L L H L H1N*
V0014 CXXXXXXXX00XNOXL L L L L L H L H1N*
V0015 CXXXXXXXX00XNOXL L L L L L H L H1N*
V0016 CXXXXXXXX00XNOXL L L L L L H H L1N*
V0017 CXXXXXXXX00XNOXL L L L L L H H L1N*
V0018 CXXXXXXXX00XNOXL L L L L L H H H1N*
V0019 CXXXXXXXX00XNOXL L L L L L H H H1N*
V0020 CXXXXXXXX00XNOXL L L L L L L L L1N*
V0021 CXXXXXXXX00XNOXL L L L L L L L H1N*
V0022 CXXXXXXXX00XNOXL L L L L L L L H1N*
V0023 CXXXXXXXX00XNOXL L L L L L L H H1N*
V0024 CXXXXXXXX00XNOXL L L L L L L H L1N*
V0025 CXXXXXXXX00XNOXL L L L L L L H L1N*

```

7-BIT COUNTER

Figure 10. ABEL JEDEC File with Test Vectors (Cont'd)

```

V0026 CXXXXXXXX00XNOXLLLHLHHL1N*
V0027 CXXXXXXXX00XNOXLLLHLHHH1N*
V0028 CXXXXXXXX00XNOXLLLHHL1L1N*
V0029 CXXXXXXXX00XNOXLLLHHL1H1N*
V0030 CXXXXXXXX00XNOXLLLHHL1L1N*
V0031 CXXXXXXXX00XNOXLLLHHLHH1N*
V0032 CXXXXXXXX00XNOXLLLHHL1L1N*
V0033 CXXXXXXXX00XNOXLLLHHHL1H1N*
V0034 CXXXXXXXX00XNOXLLLHHHHL1N*
V0035 CXXXXXXXX00XNOXLLLHHHHH1N*
V0036 CXXXXXXXX00XNOXLLHLL1L1L1N*
V0037 CXXXXXXXX00XNOXLLHLL1L1L1N*
V0038 C111111010XNOXLLHHHHHHXN*
V0039 CXXXXXXXX00XNOXLHLL1L1L1L1N*
V0040 CXXXXXXXX00XNOXLHLL1L1L1L1N*
V0041 C011111110XNOXLHHHHHHLXN*
V0042 CXXXXXXXX00XNOXHHHHHHH1N*
V0043 CXXXXXXXX00XNOXLLL1L1L1L1N*
CAD62*
D2A3

```


COPY PAL PATTERNS INTO GAL DEVICES

The generic architecture of the GAL family offers the user many different device configurations. One particular subset of these myriad architectural possibilities is common PAL architectures. GAL devices are capable of emulating all common PAL architectures with 100% pin, fuse-map, function and parametric compatibility. In short, a GAL device can drop right into any PAL socket.

This technical brief addresses the procedure of copying a PAL pattern-either a PAL master device or a PAL JEDEC file-into a GAL device. The technique is straightforward since existing files or masters can be used without modification. The first step in copying a PAL pattern into a GAL device is to determine whether the GAL device can emulate the particular architecture in question. This is accomplished by referencing tables 1 and 2.

Table 1 lists all of the 20 pin PAL architectures available as a subset of 20 pin GAL16V8 configurations.

Likewise, table 2 lists all of the 24 pin PAL architectures available as a subset of the 24 pin GAL20V8 configurations. The user must simply cross-reference the PAL architecture in question with these tables to determine which GAL device is needed.

Functional compatibility is an obvious requirement for copying to be performed. GAL devices are designed with a versatile Output Logic Macrocell (OLMC) that allows emulation of more than twenty different PAL architectures. The OLMC, shown in Figure 1, can be programmed to any of the configurations shown in Figures 2 through 5, namely: dedicated input, dedicated combinational output with programmable polarity, combinational output with feedback and programmable polarity. These four macrocell configurations can be combined as needed to allow full pin and function compatibility with the PAL architectures listed in the tables 1 and 2.

Table 1. PAL Architectures emulated by GAL16V8

GAL16V8							
10L8	12L6	14L4	16L2	16R8	16R6	16R4	16L8
10H8	12H6	14H4	16H2	16RP8	16RP6	16RP4	16H8
10P8	12P6	14P4	16P2				16P8

TABLE 2. PAL Architectures emulated by GAL20V8

GAL20V8							
14L8	16L6	18L4	20L2	20R8	20R6	20R4	20L8
14H8	16H6	18H4	20H2	20RP8	20RP6	20RP4	20H8
14P8	16P6	18P4	20P2				20P8

Figure 1. Output Logic Macrocell (OLMC): Pin 12 and 19 (GAL 16V8); Pin 15 and 22 (GAL 20V8)

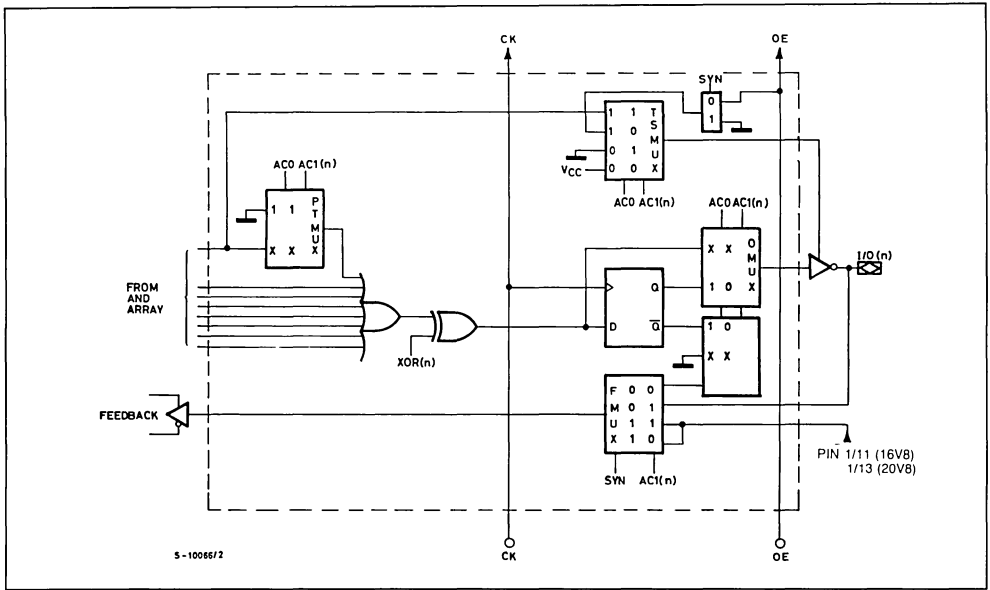


Figura 1a. Output Logic Macrocell (OLMC): Pin 13 to 18 (GAL 16V8); Pin 16 to 21 (GAL 20V8)

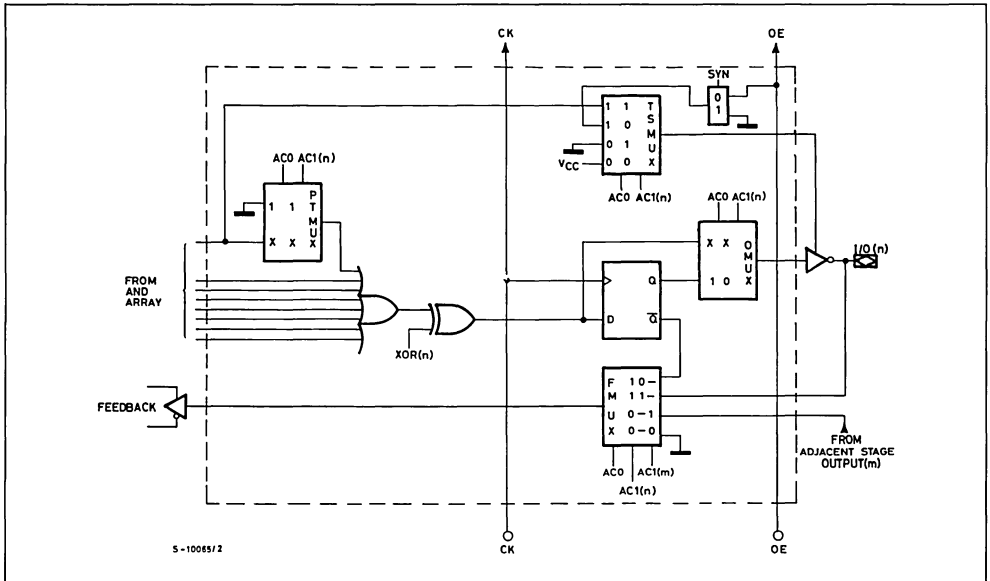


Figure 2. Dedicated Input

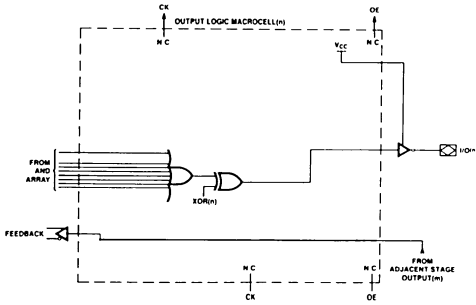


Figure 3. Dedicated Combinational Output

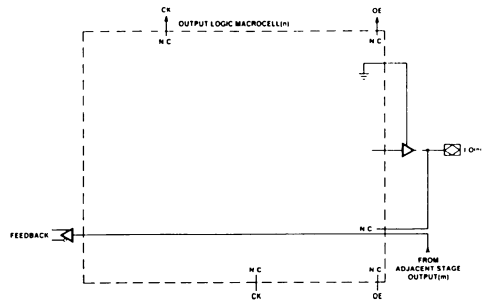


Figure 4. Combinational Output with feedback

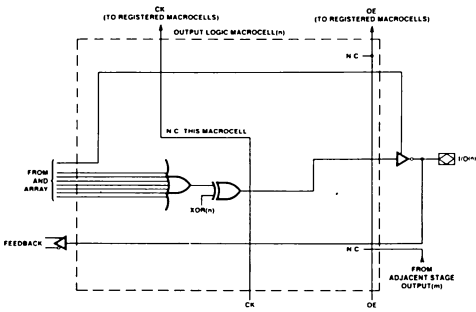
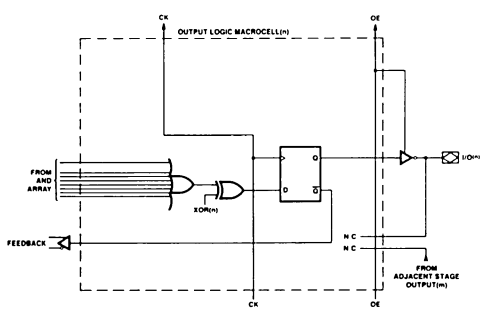


Figure 5. Registered Active-High or Low Output



Fuse-Map Compatibility

The ability of GAL devices to directly accept PAL patterns is the result of more than just functional compatibility. There is an additional stipulation that the "fuse" maps be compatible otherwise the copy procedure requires manual intervention in the form of manipulating JEDEC files or "fuse" plots to get everything into the proper format. Obviously, any manual intervention would be cumbersome and time consuming, and would significantly detract from the utility of the GAL devices.

Therefore, the GAL cell array is designed to be an identical bit-for-bit mapping of a PAL fuse array, in any output configuration. This means that external input signals and feedback signals are physically connected to the same array input lines in the GAL device as they are in a PAL device. Although this may sound trivial, consider the fol-

lowing example, which examines how different PAL architectures have different array hook-up schemes.

Refer to the partial logic diagrams of a PAL 16L2 and a PAL 16R8 in figure 6 and 7 respectively. Notice in the first figure that pin 1 is an input and is connected to array input lines 2 and 3. Now notice in the second figure that pin 1 is a clock input, and the feedback signal from pin 19 is connected to input lines 2 and 3. These differences pose no problem for the PAL manufacturer, since different PAL devices are manufactured independently of each other. However, for a GAL device to have the flexibility to accommodate different array hook-up schemes required some innovative design techniques. Although these techniques add to the complexity of the OLMC, the benefits of a directly copyable device are obvious.

Figure 6. Partial Logic Diagram of a PAL 16L2 Device

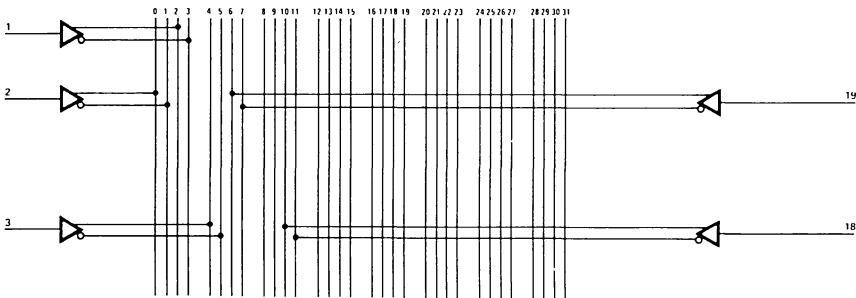
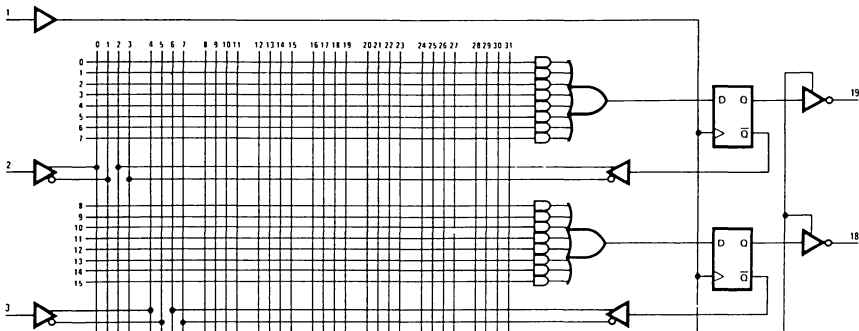


Figure 7. Partial Logic Diagram of PAL16R8 Device



GAL DEVICE APPLICATIONS

Figure 8. GAL Device Pinout Codes

DEVICE	PART TYPE	NO. OF PINS	FAMILY/PINOUT CODE	LOGICPACK REVISION	LOGICPACK ADAPTER	ADAPTER REVISION	DESIGN ADAPTER	DESIGN ADAPTER REVISION	MODEL 60A ADAPTER	MODEL 60 ADAPTER REV.	ABEL VERSION
10L8-15/-35	RAL	20	36/13	V04	303A-009	V03	303A-100	V01			
10P8-15/-35	RAL	20	36/32	V04	303A-009	V03	—	—	360A-001A	V12	2.0 or 3.0
12H6-15/-35	RAL	20	36/19	V04	303A-009	V03	303A-100	V01			
12H6-15/-35	RAL	20	36/19	V04	303A-009	V03	303A-100	V01			
12L6-15/-35	RAL	20	36/14	V04	303A-009	V03	303A-100	V01			
12P6-15/-35	RAL	20	36/33	V04	303A-009	V03	—	—	360A-001A	V12	2.0 or 3.0
14H4-15/-35	RAL	20	36/20	V04	303A-109	V03	303A-100	V01			
14H8-15/-35	RAL	24	36/08	V04	303A-009	V03	303A-100	V01			
14L4-15/-35	RAL	20	36/15	V04	303A-009	V03	303A-100	V01			
14L8-15/-35	RAL	24	36/02	V04	303A-009	V03	303A-100	V01			
14P4-15/-35	RAL	20	36/34	V04	303A-009	V03	—	—	360A-001A	V12	2.0 or 3.0
14P8-15/-35	RAL	24	36/38	V04	303A-009	V03	—	—			
16H2-15/-35	RAL	20	36/22	V04	303A-009	V03	303A-100	V01			
16H6-15/-35	RAL	24	36/09	V04	303A-009	V03	303A-100	V01	360A-001A	V12	2.0 or 3.0
16H8-15/-35	RAL	20	36/25	V04	303A-009	V03	303A-100	V01			
16L2-15/-35	RAL	20	36/16	V04	303A-009	V03	303A-100	V01			
16L6-15/-35	RAL	24	36/03	V04	303A-009	V03	303A-100	V01			
16L8-15/-35	RAL	20	36/17	V04	303A-009	V03	303A-100	V01	360A-001A	V12	2.0 or 3.0
16P2-15/-35	RAL	20	36/35	V04	303A-009	V03	—	—			
16P6-15/-35	RAL	24	36/39	V04	303A-009	V03	—	—			
16P8-15/-35	RAL	20	36/30	V04	303A-009	V03	—	—			
16R4-15/-35	RAL	20	36/81	V04	303A-009	V03	303A-100	V01	360A-001A	V12	2.0 or 3.0
16R6-15/-35	RAL	20	36/80	V04	303A-009	V03	303A-100	V01			
16R8-15/-35	RAL	20	36/82	V04	303A-009	V03	303A-100	V01			
16RP4-15/-35	RAL	20	36/85	V04	303A-009	V03	—	—			
16RP6-15/-35	RAL	20	36/86	V04	303A-009	V03	—	—	360A-001A	V12	2.0 or 3.0
16RP8-15/-35	RAL	20	36/87	V04	303A-009	V03	—	—			
16V8-15/-35	GAL	20	36/55	V04	303A-009	V06	—	—			
18H4-15/-35	RAL	24	36/10	V04	303A-009	V03	303A-100	V01			
18L4-15/-35	RAL	24	36/04	V04	303A-009	V03	303A-100	V01	360A-001A	V12	2.0 or 3.0
18P4-15/-35	RAL	24	36/40	V04	303A-009	V03	—	—			
20H2-15/-35	RAL	24	36/11	V04	303A-009	V03	303A-100	V01			
20H8-15/-35	RAL	24	36/61	V04	303A-009	V03	303A-100	V01			
20L2-15/-35	RAL	24	36/05	V04	303A-009	V03	303A-100	V01	360A-001A	V12	2.0 or 3.0
20L8-15/-35	RAL	24	36/26	V04	303A-009	V03	303A-100	V01			
20P2-15/-35	RAL	24	36/41	V04	303A-009	V03	—	—			
20P8-15/-35	RAL	24	36/62	V04	303A-009	V03	—	—			
20R4-15/-35	RAL	24	36/65	V04	303A-009	V03	303A-100	V01	360A-001A	V12	2.0 or 3.0
20R6-15/-35	RAL	24	36/66	V04	303A-009	V03	303A-100	V01			
20R8-15/-35	RAL	24	36/27	V04	303A-009	V03	303A-100	V01			
20RP4-15/-35	RAL	24	36/46	V04	303A-009	V03	—	—			
20RP6-15/-35	RAL	24	36/64	V04	303A-009	V03	—	—	360A-001A	V12	2.0 or 3.0
20RP8-15/-35	RAL	24	36/63	V04	303A-009	V03	—	—			
20V8-15/-35	GAL	24	36/57	V06	303A-009	V06	—	—			

The "••" symbol means that the algorithm for this device is under evaluation for possible addition in some future update.

Programmers with RAL Device Codes

The final element required to facilitate direct PAL device copying is the RAL (Reprogrammable Array Logic) code. The RAL code is the user's means of informing the programmer box exactly which architecture the GAL device will be patterned to. It correlates to the family/pin code selection for identifying PAL architectures.

The information shown in figure 8 is taken from the DATA I/O wall chart listing all device manufacturers' family pin codes.

Notice that there is an entry for each GAL of the GAL16V8 and the GAL20V8, as well as for many RAL codes. It should be noted that, although there are only two physically different device types - the GAL16V8 and GAL20V8 - these products source all of the 44 device types listed in the figure 8; the RAL codes are provided merely for the purpose of copying PAL devices. To emulate a PAL16R4 family/pin code must be used according to the procedure explained below. The RAL codes directly correspond to each respective PAL device configuration and provide the programmer box with the information to program the GAL device OLMCs. Physically resident inside the GAL device is an 82 bit architecture control word which, once programmed, determines the configuration of the OLMCs. (Refer to the data-sheet for further discussion on the architecture control word). Downloading a PAL pattern to a programmer box transfer array data only; no information as to the device architecture is transmitted. Thus the RAL code is essential.

Inside the memory of a device programmer is the predetermined GAL architecture control word corresponding to each PAL device configuration. By selecting the appropriate RAL code, the device programmer can append the predetermined architecture control word to the array information previously downloaded. It stands to reason that if the GAL family/pin code were selected, the GAL device's array would be programmed properly but the OLMCs would be left unprogrammed, since there is no predetermined OLMC configuration in a virgin GAL device.

The GAL family/pin code would only be used in the development of a new design from scratch. In the JEDEC file generated by the development software is contained an architecture control word specific to the design in question. When copying an existing PAL master device or an existing PAL JEDEC file, the architecture control word only gets physically appended to the cell array information by the programmer box when the appropriate RAL code is selected.

A final reminder: When copying PAL device patterns into GAL devices, use the RAL family/pin code. When beginning a GAL design from scratch, use the GAL family/pin code.

Actual steps for copying PAL service patterns into GAL devices using programmers

With-RAL codes

This procedure is generic in nature and is not specific to any device programmer. A PAL master device or a PAL JEDEC file may be used.

- 1 - Load either the PAL-device fuse map or JEDEC file data into the device programmer memory using the normal procedure from.
- 2 - Select the appropriate RAL family/pin code form the programmer chart. Note: Remember that the RAL code is required for PAL device copying in order to configure the OLMCs properly. Do not use the GAL device code for copying PAL devices.

(Note: Some programmers require changing adapters when switching from one device manufacturer to another)

- 3 - Program the GAL device using the appropriate RAL family/pin code to configure to OLMCs properly. The copy procedure is complete and resulting GAL device is 100% compatible with PAL device it copied.

(Note: The GAL device still retains its full erasability feature. The device can be reused with different array patterns and architecture configurations (RAL codes) as selected by the designer).

PAL to GAL JEDEC File Conversions

GAL devices are fuse map compatible with standard PAL devices, so any development software that is capable of creating JEDEC files for use with PAL devices, such as PALASM, can be used with GAL devices. However, GAL devices have additional "fuses" that define their architecture (synchronous, asynchronous, or combinations thereof). This architecture information is not provided by PAL only assemblers and compilers. Most programming hardware configure GAL device architectures automatically, using menus or other means to determine the device being emulated. However, some programming fixtures do not configure GAL device architectures automatically, requiring instead, that all necessary architecture information be present in the JEDEC file. This means that JEDEC files developed for use with PAL devices must be modified before they can be used with GAL devices on such a programmer. The program PALtoGAL performs all of the necessary translations required. The program PALtoGAL, which is distributed by SGS-THOMSON, is written in turbo Pascal 3.01 A;

it supports both 16V8 and 20V8 GAL devices. The typical environment used for this software could be an IBM personal computer or equivalent machines, with the following minimum configuration:

- * Single floppy disk drive 5 1/4" – 360KB or 3 1/2" – 720KB
- * MS-DOS or PC-DOS operating system, 2.0 or later
- * 256KB Minimum free RAM size

The modification PALtoGAL performs are as follows:

1) Appends the proper architecture information.

- 2) Sets unused product terms to all zeros.
- 3) Sets unused input terms to all ones.
- 4) Expands the fuse map to include every fuse (recognizes, but doesn't use the F field).
- 5) Includes a fuse address (L field) every 32 fuses (every 40 fuses for GAL20V8) for JEDEC file readability.
- 6) Recalculates fuse array and transmission checksums.
- 7) Adds "QP" (quantity of pins) and "QF" (quantity of fuses) fields.
- 8) Insert user electronic signature, if specified.

USING ELECTRONIC SIGNATURE

In the course of system development and production, the proliferation of PLD architectures and patterns can be great. To further complicate the record-keeping process, design changes often occur, especially in the early stages of product development. The task of maintaining "which pattern goes to what device for which socket" becomes exceedingly nontrivial. What's more, once a manufacturing flow has been set, it becomes important to "label" each PLD with pertinent manufacturing information, which can be quite beneficial in the event of the customer return traceability aided by a manufacturing history can help to quickly reconstruct details of a defective product and thereby effect a speedy solution. The SGS-THOMSON GAL family can ease the problems associated with document control and traceability, thanks to a feature called Electronic Signature (ES). This brief describes the concept behind ES, how it is used, and the advantages obtainable with regard to manufacturing-flow control, documentation, and traceability.

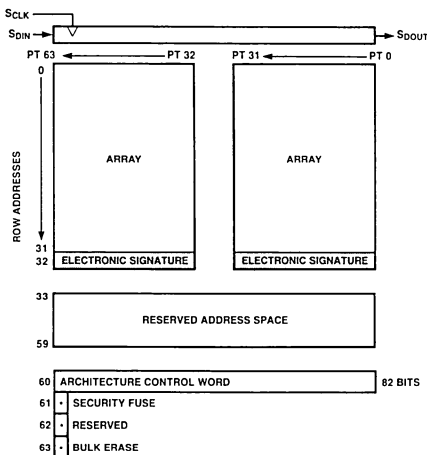
Electronic Signature is basically a user's "notepad" provided in electrically erasable (EE) cells on each GAL device. Essentially an extra row that's appended to the array and allocated for data storage; the ES can contain up to 8 bytes. Figure 1 shows Row Address Maps for the GAL 16V8 and GAL 20V8, with the ES row highlighted.

Because one GAL device can emulate many different PAL architectures, inventory logistics are greatly simplified. To further simplify the development and manufacturing process, SGS-THOMSON incorporated ES to store such manufacturing data as the manufacturer's ID, programming date, programmer make, pattern code, revision number, and product flow. The intent was to assist users with the complex chore of record maintenance and product flow control. In practice, the ES can be used for any of a number of ID functions.

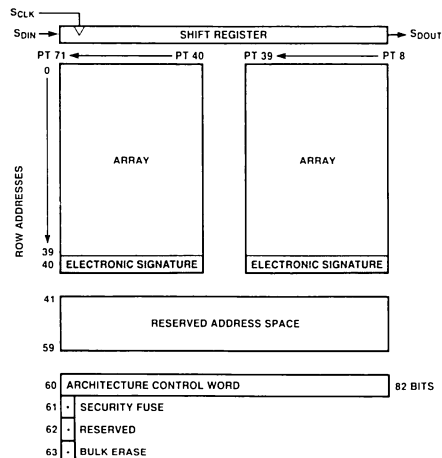
For user simplicity, ES will be supported by all GAL family software support packages such as CUPL and ABEL, which are discussed with detail in section 4 of this handbook. The user will be able to define data fields, specify information, and write or read ES data from each GAL device. The following paragraphs describe how ES may be managed.

Within the 64 bits (eight bytes) available for ES data storage, users may find it helpful to define specific fields, to make better use of information storage. A field may use only one bit (or all 64), and may contain a variety of topics. Some fields should probably be reserved for future expansion. The possibilities for fields are endless, and completely up to the user. As an example, figure 2 divides the ES in five fields: manufacturer's ID, device program date, programmer ID code, pattern code, and reserved section.

Figure 1. Row Address Maps



GAL16V8



GAL20V8

Even with the GAL device's security feature enabled, the ES can still be read. If a pattern code were stored in the ES, the user could always identify which pattern had been used in a given device. In this way, a device pattern could be confidentially retrieved. As a second safety feature, when a GAL device is erased and repatterned, the ES row is automatically erased. This prevents any situation in which an old ES might be fitted with a new pattern. (No information is better than wrong information). It is the user's responsibility to update the ES when reprogramming.

Programming the ES is accomplished in the same manner as any other array write operation. With the GAL device in Edit mode, the ES can be selected using a row address of 32 on the GAL 16V8, or 40 on the GAL 20V8. By following the shift register and program timings (specified in the data sheets of section 2) to load, program and verify the ES, a routine can be developed on most ATE systems to pattern a user-specific signature into the device.

Though provided to assist the designers and manufacturers who utilize GAL products, making use of ES is not essential to enjoying the many benefits of GAL devices. For those willing to invest in it, however, the reduction of "hidden costs" associated with PLDs can be significant.

Eliminating labels

By automatically storing the appropriate identification information into GAL device ES locations while the programming hardware is patterning the device, the need for a costly additional handling step to apply messy gummed labels or ink is eliminated. What's more, throughput and quality of the patterned devices is greatly increased.

Document Control

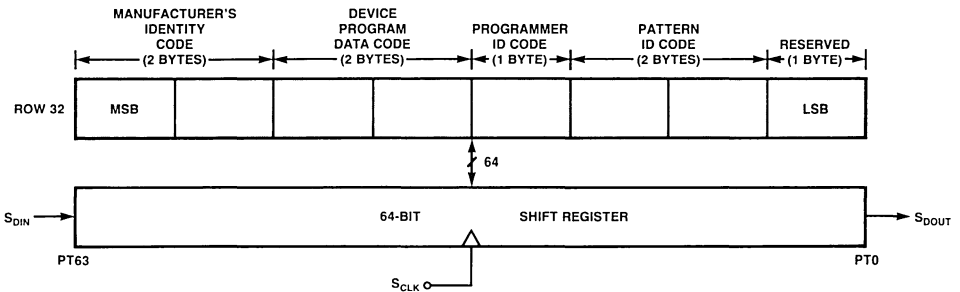
The job of document control becomes more manageable when using the GAL device ES, since a pattern code in the ES can specify each pattern and its application. This proves an absolute boon in military programs, where accurate documentation is essential. If a change occurs, it is easily handled with a pattern code in each device, a readout can actually be conducted during board assembly. Code verification would ensure the use of properly patterned devices and serve as a quality monitor step. Moreover, validation is simplified when checking against a lot-or board-traveler, since master devices are not required.

Software Revisions

With ES, a software ID code can be stored and referenced in Document Control to a current-pattern version. When a revision occurs, a new pattern code is simultaneously stored in the ES. For the first time in PLD history, pattern codes can be monitored to verify that incorrect versions of software are not inadvertently being used.

With GAL devices, of course, any material flagged with an improper pattern code can simply be sent back and reprogrammed to the current-pattern revision. Also, when security is enabled, an ES-resident pattern ID code is the only certain means of documenting which pattern resides within a device.

Figure 2. Typical ES Field Definitions



Manufacturing Information

As described earlier, manufacturing information stored in the GAL device ES can help track down problems, should products be returned. If each board-assembly location were coded into GAL devices used as that assembly site, for example, customer board returns might be linked to a common source. Also, identification codes would eliminate the need to use external labels or stamps to signify different vendors.

Manufacturing Flow

With ES, devices can all be preprogrammed at one location and given a destination code. Upon shipment and receipt, sample readouts of destination codes could be performed to ensure that the proper devices were received. As systems become more complex, production and document control costs can become dominant. Electronic Signature is one of many valuable ease-of-use features offered in the GAL device family that can tame such costs. SGS-THOMSON will continue to deliver outstanding user support, by making the ES feature available on all GAL devices.

OVERVIEW OF REGISTER PRELOAD

Register preload is a testability feature that is rapidly gaining prominence throughout the chip-design market place. Because it allows any arbitrary state value to be loaded into a PLD's output register, this powerful feature is capable of breaking down complex logic design into simple testable blocks. All GAL family devices from SGS-THOMSON incorporate the feature, and though extremely valuable for testing the device during manufacture, register preload allows the discerning user to convince himself of this bold claim: because of the testability advantages of the E²CMOS technology, SGS-THOMSON guarantees 100% programming and functional yields for GAL devices after programming. Although its comprehension is not a prerequisite for GAL device usage, this technical brief provides a technical overview of register preload and how it can be useful.

By allowing any arbitrary state value to be loaded into a device's output registers, register preload can provide an easy method of testing registered devices for logical functionality.

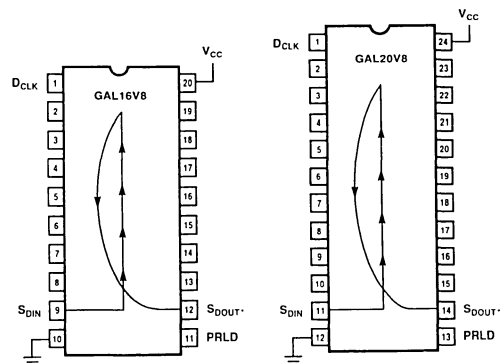
For conclusive testing of state-machine designs, all possible states and state transitions must be verified, not just those required in the normal machine operations.

This is because, during system operation, certain events (power-up, line voltage glitches, brown-outs, etc.) can throw the logic into an illegal state. To test a design for proper handling of such conditions,

a way must be provided to break the feedback paths, and force any desired - even illegal - state into the registers. Then the machine can be sequenced, and the outputs tested for the correct next-state conditions.

The GAL 16V8 and GAL 20V8 devices include circuitry that allows each registered output to be synchronously set either high or low. Thus, any present-state-condition can be forced for test sequencing. Figure 3 shows the pin functions necessary to preload the registers. The register preload timing and pin voltage levels necessary to perform the function are shown in figure 4 (See section GAL device specification) for parametric specifications and the register preload. This test mode is entered by raising PRLD to V_{IES} (a supervoltage of 15V), which enables serial data in (S_{DIN}) buffer and the serial data out (S_{DOUT}) buffer. Data is serially shifted into the registers on each rising edge of the clock, DCLK. Only the macrocells with registered output configurations are loaded. If only 3 outputs have registers, then only 3 bits need to be shifted in. The registers are loaded from the bottom up, as illustrated in figure 3. A typical functional test sequence would be to verify all possible state transitions for the device being tested. To verify these transitions requires the ability to set the state registers into an arbitrary "present-state" value, and to set the device inputs to any arbitrary "present-input" value. Once this is done, the state machine is then clocked into a new state or "next-state". The next state is then checked to validate the transition from the present state. In this way any state transition can be checked.

Figure 3. Output Register Preload Pin Configuration



* The S_{DOUT} - output buffer is an open-drain output. This pin should be terminated to V_{CC} with a 10K resistor.

Shorter test sequences

The difficulty in getting to certain states or conditions can lead to logic-verification sequences that are either incomplete or excessively long. Long test sequences result when feedback signals from state registers combine with inputs to determine the "next-state" values. This condition forces the state machine to go through many state transitions before it can reach the state that requires testing. Therefore, the test sequence will be mostly state initialization and not actual testing. The test sequence can become excessively long when a state must be reentered many times to test a wide variety of input combinations.

Consider a device programmed as a seven-bit counter that, among other functions, asserts an output signal from an eighth output once the counter reaches 127; or binary 1111111. In order to test this eighth output for the proper logic level; it is necessary to cycle the counter 127 times just to set up the desired test condition. Also if several different input combinations need testing with seven bit counter in this state, each set of input conditions will require the test sequence to lengthen by 128 test vectors, of which 127 vectors are for initialization only.

Register preload allows the desired "present-state" to be loaded into the device in one test vector and allows testing for the "next-state" with a second test vector. The benefits of this feature are readily apparent for shortening test sequences.

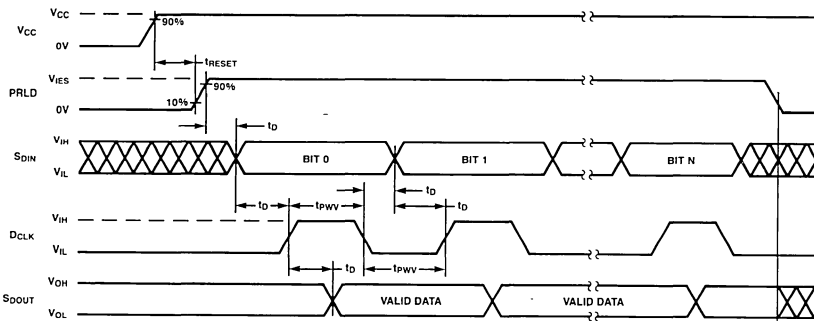
Unreachable States

Complete logic verification is often impossible when states that need to be tested can not be entered with normal state transitions. Included among these are "forbidden", "power-up", or "don't care" states that are not normally entered but need to be tested to ensure that they return to the defined state. Consider a design including a counter that counts the sequence 0,1,2,5,0,1,2,5,... Under normal conditions, the state registers for this counter would never reach states "3" and "4". However, to guarantee proper functionality, these states must be entered, to ensure the machine will reset if either of these forbidden states is ever errantly entered. Without register preload, this task is impossible, since there would be no way to force the counter into forbidden states. With register preload, the states are simply loaded into the counter, which is subsequently clocked to the next state to verify proper return to a known state and normal operation.

For the most demanding users, register preload is an invaluable feature for testing all possible state transitions in a registered device, and one that can significantly shorten test sequences, as well as allow forbidden states to be tested.

These benefits translate to higher quality systems, because of the greater degree of confidence that all components in the system function properly prior to shipment. The point for all users of GAL devices to remember is that the inherent testability of SGS-THOMSON's E²CMOS technology removes the testing burden from the user by assuring the highest-quality programmable logic devices available 100% programming yield, 100% functional yields guaranteed.

Figure 4. Timing Diagram

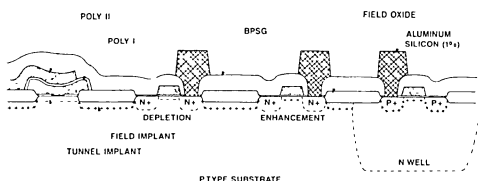


TECHNOLOGY AND TECHNICAL OVERVIEW

E²CMOS Technology and Circuit Considerations

The E²CMOS technology employed on all GAL devices combines a state-of-the-art, high-performance CMOS static RAM process with an electrically erasable, nonvolatile memory process to form SGS-THOMSON's E²CMOS technology. This technology yields products with the best speed-power characteristics available from any user-programmable technology. A cross section of the E²CMOS technology is illustrated in figure 1, with key process features listed in table 1. The foundation of the technology is an oxide-isolated N-well CMOS process fabricated on a P-type substrate.

Figure 1. E²CMOS Process Cross-Section



further reduce device capacitance in all areas. The E²CMOS technology utilizes fully self-aligned, polysilicon-gate processing coupled with low temperature process steps to minimize key circuit and device performance limiting parasitic capacitances, such as gate-source and gate-drain overlap capacitances.

The floating-gate memory cell technology utilized is based upon a 95 Å oxide, utilizing the principles of Fowler-Nordheim tunnelling. This technology was chosen due to its high quality, reliability and repeatability.

In fact, the memory cell proliferated throughout the GAL family of devices has proven its reliability through exhaustive testing (millions of write-erase cycles) and over extended periods of time (over 1,000,000 device-hours have been logged at the time of this writing).

In addition to the floating-gate (or "sense") device, an additional select transistor (or "pass gate") is added in series with the cell in E²CMOS designs, to isolate the sense device from the array during read and write operations. A schematic representation of this type of cell is shown in figure 2. In addition to the conventional bit line and word line, the E²CMOS cell has additional line for control gate, or top gate, which controls the potential of the floating gate. In the GAL family of programmable logic devices, this line occupies little additional area.

Figure 2. Typical E²CMOS Cell

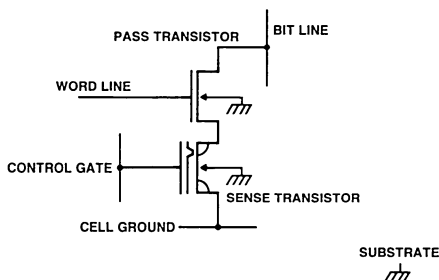


Table 1. E²CMOS Dimensional Features

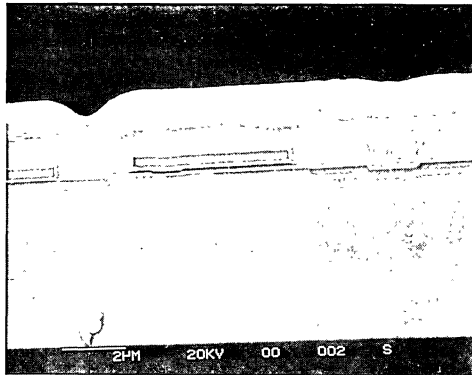
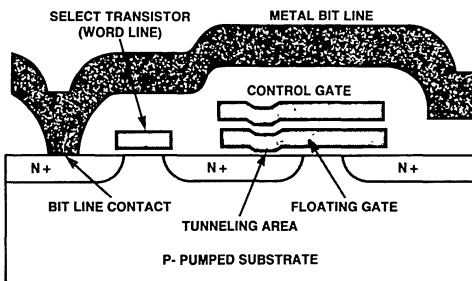
P-Type Substrate (20Ω - cm)
N-Well Bulk Silicon CMOS
Double-Level Polysilicon
Single-Layer Metalization
1.0 μm Channel Length
OXIDES
500 Å First Gate
330 Å Second Gate
420 Å Interpoly Oxide
95 Å Tunnel Oxide
KEY PITCHES UTILIZED
5.2 μm Active Area
4.0 μm Second Poly
4.8 μm Metal

The use of negative substrate-bias generation eliminates the requirements for expensive (and defect-inducing) epitaxial processing techniques, while enhancing device performance due to reduced junction capacitance on all N-channel devices. The oxide-isolation techniques serves to

The cell is programmed by applying a programming pulse to either the control gate or the drain region of a cell that has been selected by an applied high voltage on the select-line.

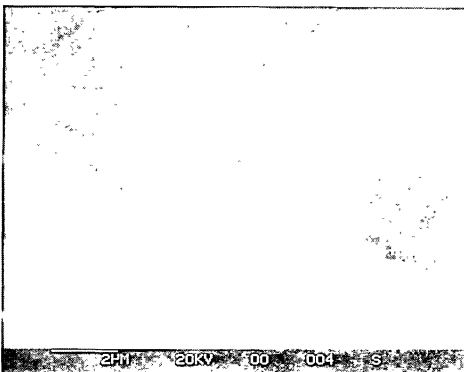
Programming takes place when electrons "tunnel" through a thin dielectric, which is indicated in the schematic by the small notch in the floating gate over the drain of the sense device. A microphotographed cross-section of the E²CMOS cell, as well

Figure 3. E²CMOS Cell Cross-Section



as its schematic representation, is shown in figure 3. Note that the sense-transistor channel length is defined by a masking layer and not by polysilicon gate, as in the case of the select, or pass, transistor. This technique minimizes the size of the cell while maintaining a high coupling ratio between the floating gate and the control gate. As in the case of an ultraviolet-light erasable PROM (EPROM) cell, the E²CMOS cell requires a very low-leakage, high-quality oxide between the two levels of polysilicon to guarantee good data retention characteristics. A more dramatic illustration of the cell is shown in the scanning-electron-microscope (SEM) photograph of figure 4 in which the various layers have been etched back to reveal the floating-gate structure. Once the charge has been placed on the floating gate, the actual floating-gate potential can be modulated by the voltage on the control gate

Figure 4. Etched Back E²CMOS Cell



through capacitive coupling. It is this capacitive coupling that is used to generate the high voltage across the tunnel dielectric at the beginning of a programming pulse. During a programming cycle, the cell is first erased to a 1, or nonconducting state, then selectively written to a 0, or conducting state by a write cycle. This prevents the sense device from conducting current during the write operation, when voltage is applied to the drain of the device.

E²CMOS Circuitry

The circuitry employed on all GAL devices has been specifically designed to provide user key benefits. AC and DC parametric performance has been optimized over the full military temperature and voltage range, through the use of substrate-bias generation coupled with temperature compensated, self-biasing sense amplifiers.

All GAL devices have been designed to provide total parametric compatibility with existing bipolar TTL circuit technologies. Fig. 5 illustrates the typical room-temperature transfer characteristics of a number of TTL circuit technologies, along with the typical GAL device characteristic. The low-temperature and high-temperature characteristics are provided in Figures 6 and 7. The effects of temperature and voltage on two key AC parameters (input-to output and clock-to output delays) can be studied in figures 8,9,10, and 11. The scales have been normalized to 25°C and 5.0V operation, for reference.

The advantages of E²CMOS technology become apparent upon examination of these figures. The transfer characteristic is significantly improved with E²CMOS, both in terms of static V_{OH} level (at specified I_{OH}) as well as the "squareness" and narrow transition region of the curve. The switching threshold is extremely stable, as compared with

that of TTL circuit technologies, whose thresholds are based upon PN diode stacks; the latter has a strong exponential dependence on temperature, while E²CMOS thresholds are primarily a function of device size ratios, which are, of course, not temperature-dependent. The normalized access times vs. temperature

curves illustrate the linearity of E²CMOS, as contrasted with the classical bipolar "bowl" shaped temperature characteristic. It can be seen that E²CMOS produces performance commensurate with TTL above 25°C, while significantly outperforming TTL characteristics below 25°C.

Figure 5. Room Temperature Transfer Characteristics

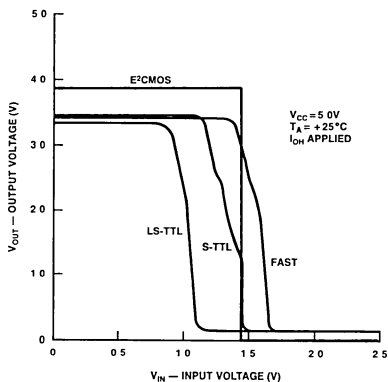


Figure 6. +125°C Transfer Characteristics

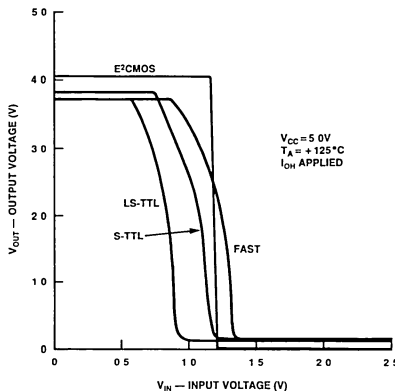


Figure 7. -55°C Transfer Characteristics

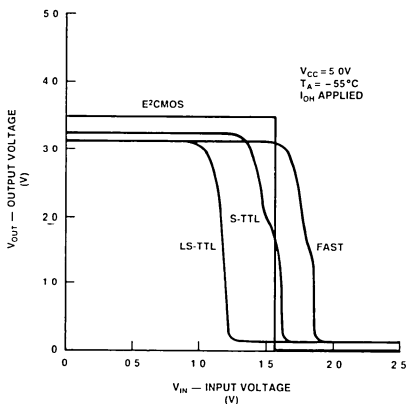


Figure 8. Normalized t_{PD} vs. Supply Voltage

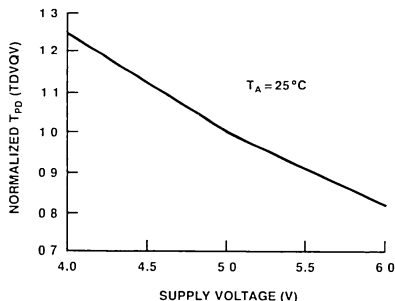


Figure 9. Normalized t_{CO} vs. Supply Voltage

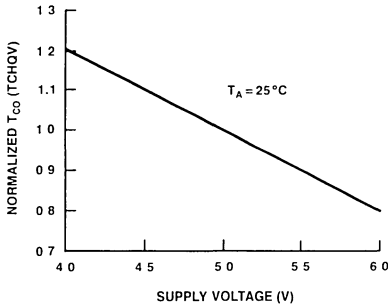


Figure 10. Normalized t_{CO} vs. Temperature

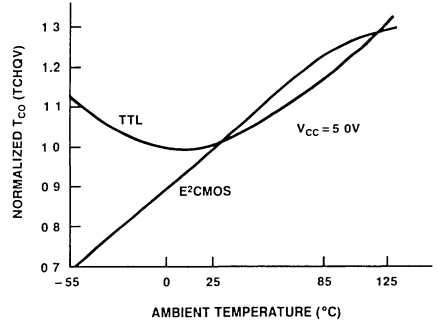


Figure 11. Normalized t_{PD} vs. Temperature

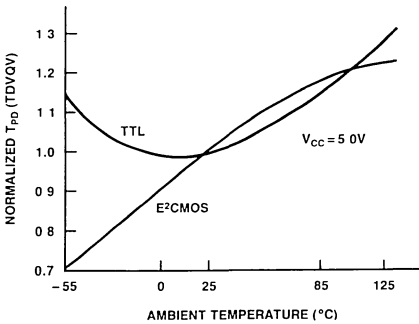
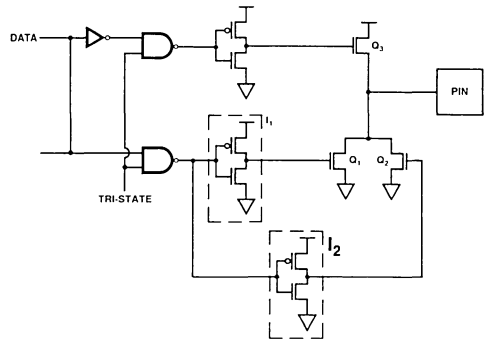


Figure 12. Phased Output Turn-On Circuit



E²CMOS Output Characteristics

The schematic representation of the output driver utilized on all GAL devices is illustrated in Figure 12. This is a specially designed high-drive output stage with two key benefits for the system designer. The first point to notice is the exclusive use of N-channel transistors for both pull up and pulldown devices in the push-pull stage connected to the device pin (Q1,Q2,Q3). The use of P-channel devices on circuit outputs is one of the major sources of destructive “SCR latch-up” common to many CMOS technologies.

The reason for this is due to output overshoot, un-

dershoot, or “ringing” due to signal reflections and noise in a system application.

The second salient feature of the GAL output-stage design is the phased, “soft” turn-on of its high-current (24mA) pull-down drivers, Q₁ and Q₂. This feature benefits system designers by controlling transient current spikes, which often occur when switching heavy, capacitive loads and many outputs in parallel. Dynamic switching spikes often play havoc with system V_{SS} and V_{CC} bus lines, due to inductive coupling according to the equation.

$$V_{NOISE} = L_{bus} \left(\frac{di}{dt} \right)$$

Where V_{NOISE} is the equivalent noise voltage, and L_{BUS} is the bus inductance. The phased output turn-on circuit serves to limit the V_{NOISE} component by effectively reducing the dl/dt term.

This is accomplished through the use of paired pull-down devices.

In effect, the circuit has two output drivers, a dynamic driver (Q_1) and a static driver (Q_2). Q_1 is driven by a very fast inverter stage (I_1), while Q_2 is driven by an inverter stage (I_2) designed to turn on more slowly (by about 1.5 ns), but still turn off

turn off quickly. This design effectively smoothes the dynamic output current spike generated when switching heavy, capacitive loads, yet provides a solid DC I_{OL} drive characteristic.

The propagation delay induced by capacitive output loading is illustrated in Figure 13 and 14. The DC output drive characteristics (I_{OH} and I_{OL}) are illustrated in figure 15 and 17. The 200-ohm load line in Figure 15 clearly illustrates the ability of GAL devices to drive heavy loads and still maintain a solid V_{OH} level of approximately 2.9 volts.

Figure 13. Delta t_{pD} vs. Output Loading

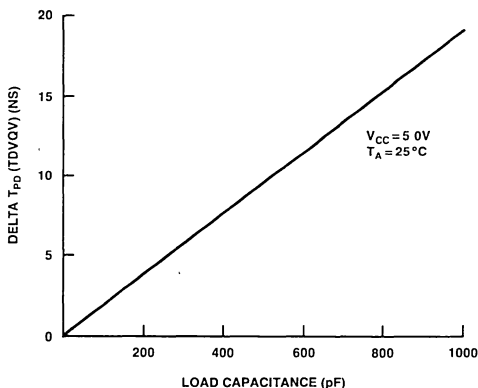


Figure 14. Delta t_{cO} vs. Output Loading

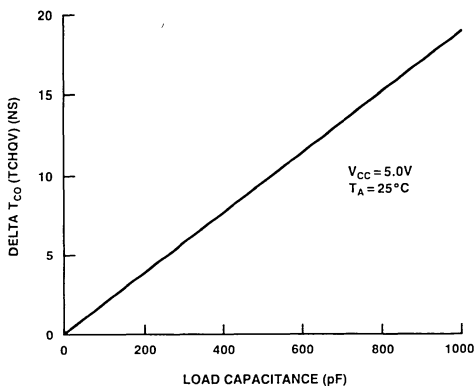


Figure 15. I_{OH} vs. Output Voltage

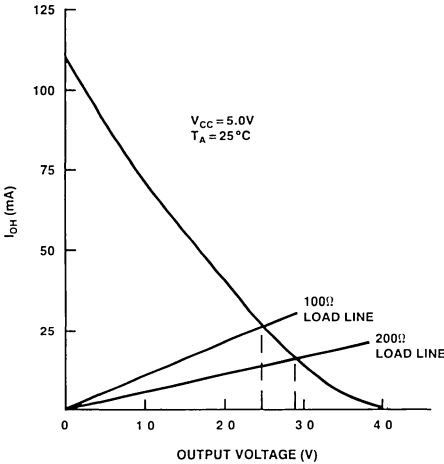


Figure 17. I_{OL} vs. Output Voltage

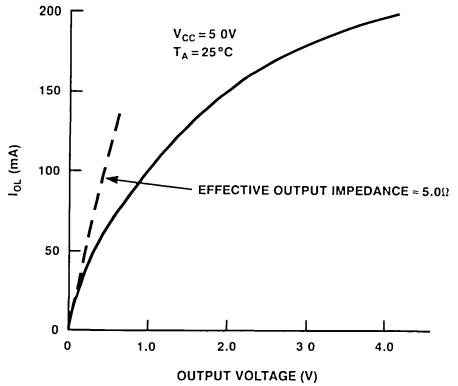
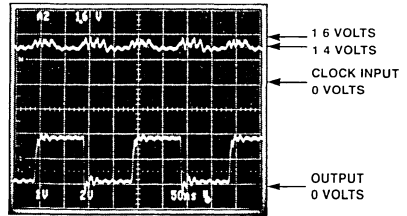
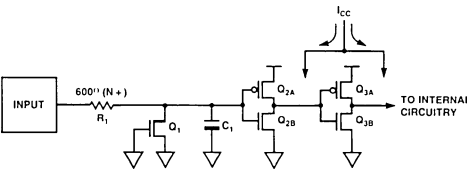


Figure 18. 10MHz Clocking with 200mV P-P Input

Figure 16. Input Translator/Buffer



E²CMOS Input Characteristics

A schematic representation of the input transistor/buffer utilized on all GAL devices is illustrated on Figure 16. The transistor is specifically designed for compatibility with existing bipolar TTL circuit technologies, while providing significantly improved noise margins. Note the presence of an internal decoupling capacitor (C_1) physically integrated on each device input and I/O. Decoupling significantly improves transient-noise rejection ratio and makes it possible to use GAL devices successfully within noisy system environments, which would induce failures in other circuit technologies.

Figure 18 is provided as an example of this "designed-in" noise margin by illustrating a GAL16V8 operating as a counter with a 10MHz clock input signal that has only a 200mV swing. This is, of course, not recommended design prac-

tive, but demonstrates the sensitivity and noise filtering capability of the GAL device input buffer. Two other important input characteristics of GAL devices are related to current-voltage (I-V) relationships. One of these is the input current (I_{IN}) versus the input voltage (V_{IN}), and is illustrated along with the common TTL characteristic in Figure 19. As can be readily discerned from the figure, E²CMOS provides the user with an ideal "open circuit" input load characteristic, whereas the TTL devices present a significant DC load to the driving circuitry. Ten TTL devices in parallel can easily present a 4mA DC load to a driver. The remaining I-V characteristic is related to the static I_{CC} consumed by the input translators as a function of the applied input voltage. As the input voltage transitions from a logic 0 to a logic 1, there exists a region typically between $V_{IN} = 1$ to 3V, in which Q_{2A} and Q_{2B} are

both slightly on and a current path exists between V_{CC} and V_{SS} . The same situation occurs with Q_{3A} and Q_{3B} , such that an E²CMOS input buffer characteristic will allow small amounts of DC current to flow as a function of input driving voltage levels. This input buffer characteristic is shown in Figure 20. Note that there are two different curves illustrated in this figure -one for dedicated input buffers, and one for I/O (bidirectional) buffers. The I/O buffers consume roughly twice as much I_{CC} as dedicated inputs, due to the fact that they have been scaled up in size to help compensate for the extra delay (about 1ns) they incur in the I/O feedback multiplexer circuit that dedicated inputs do not experience.

ESD Protection Network

It should be noted that all GAL device inputs have an integral electrostatic discharge (ESD) protection structure comprising a diffused N + resistor and a "grounded gate" transistor (refer to Figure 16) with a special phosphorous implant in its drain region that is designed to non-destructively break down between source and drain at approximately 23 volts.

This breakdown of Q_1 provides a low impedance path to ground, thus dissipating any potentially damaging charge that would otherwise pass to the internal circuitry.

GAL device inputs are guaranteed to provide in excess of 1500V of protection when tested according to the MIL-STD -38510 ESD test methods.

Figure 19. I_{IN} vs. V_{IN}

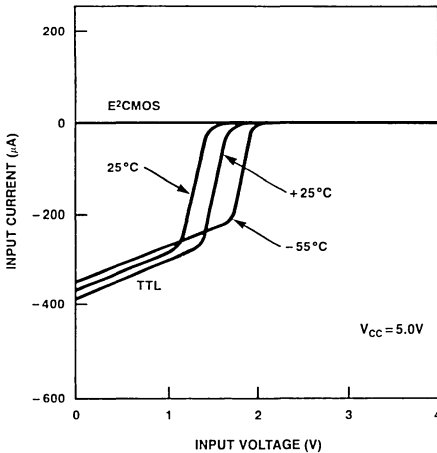
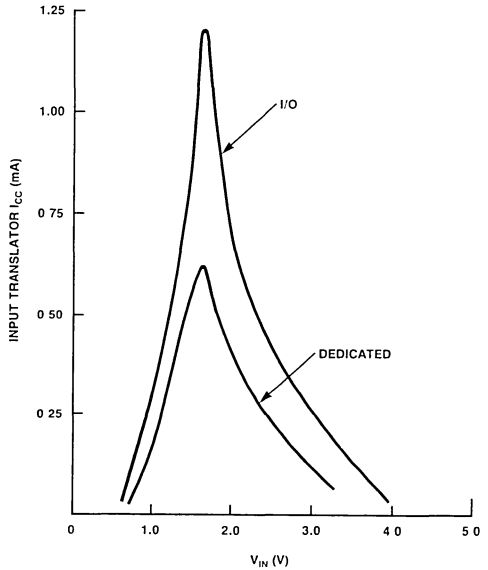


Figure 20. Input Translator I_{CC} vs. V_{IN}



Overshoot Characteristics

With signal propagation delays decreasing as a result of device improvements, signal slew rates are approaching 5V/ns in some technologies. Proper attention to impedance-matching and termination becomes paramount, as system designers are faced with transmission line effects, such as reflections which are the source of a majority of signal overshoots and undershoots on a board. Bipolar TTL circuit technologies have historically been very forgiving in this area, due to the integration of input and output Schottky "clamping" diodes.

MOS technologies, in general, do not have the luxury of having a readily available Schottky diode in the process. E²CMOS technology is no exception; to ensure that GAL devices did not have or cause problems in a customer's system, SGS-THOMSON design goals included a requirement to sustain large overshoots and undershoots on device pins while having no functional effect on the device performance. The reader is referred to Figure 21 which illustrates the same device and pattern as utilized

in Figure 18, except the clock signal is now 16 volts peak-to-peak (-8.0V to +8.0V), still clocking the circuit at 10 MHz with no device malfunction. The specific I-V curves for input and output circuits are shown in Figures 22 and 23. The output circuit characteristic is very similar to that of TTL technologies. The input circuit can be seen to have a limited current-sourcing ability, as compared with TTL technology.

Figure 21. 10MHz Clocking with 16V P-P Input

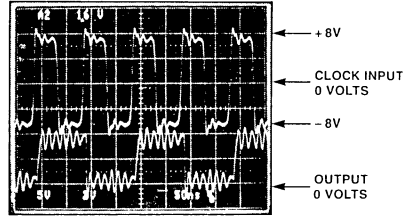


Figure 22. Output Voltage vs. Output Current

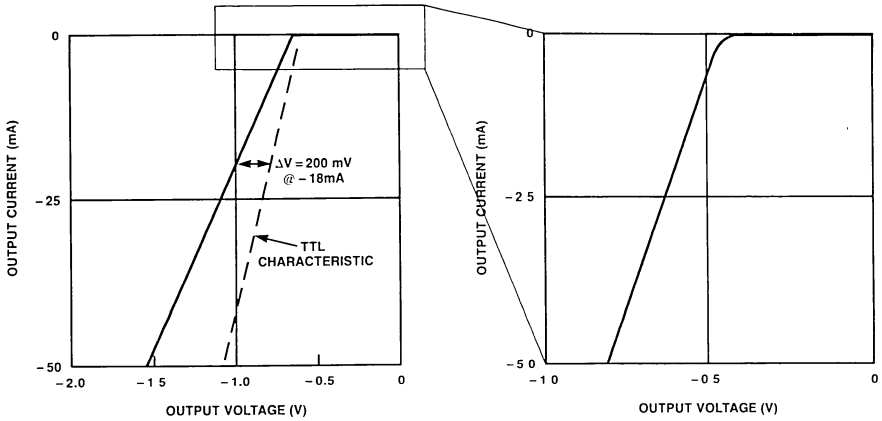
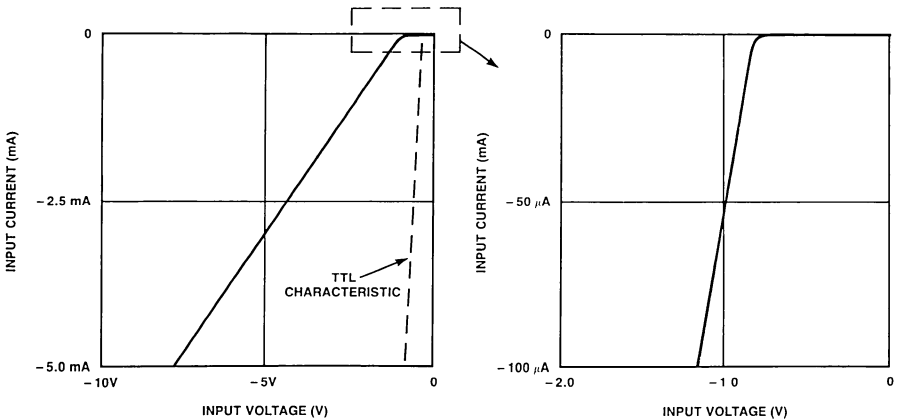


Figure 23. Input Voltage vs. Input Current



THE GAL PROCESS

While it is the responsibility of component manufacturers to maximize product quality and reliability by testing to weed out infant mortality and thereby delivering units with a low failure rate, programmable-logic-device (PLD) manufacturers have traditionally placed the responsibility of testing on the user.

Primarily because bipolar fuse-link technology does not lend itself well to testability prior to programming, PAL device consumers have been forced to bear the burden of testing.

GAL devices are the only PLDs that are completely tested before shipment.

Incorporating design-for-testability features, GAL devices are reprogrammed in milliseconds and tested extensively through the actual circuitry used in device operation, rather than by means of "shadow" arrays and "dummy" columns that one-time-programmable PLDs resort to. When a consumer purchases a GAL device and programs it, he can feel 100% confident that the device will adhere to all performance specifications outlined in the GAL data sheet. This can be assured because prior to shipment, each GAL device has been programmed with hundreds of worst-case patterns, cycled for endurance and tested over temperature with every scheme of architecture.

PLDs have historically been impossible to test and characterize on a volume-production level, because complete performance testing could not be accomplished until after a device was programmed for a specific application. Only the customer could verify that performance specifications were indeed being met, by testing each device fully after programming.

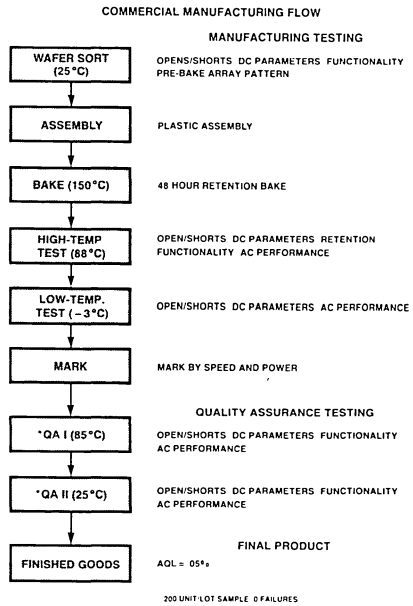
Manufacturers of bipolar fuse-link PAL devices present statistical data that implies that if the device programs, it will probably operate adequately. Consequently, the user has to program the device and test it over temperature to screen out statistical anomalies. If programming yield is 98% and functional yield is 99%, then each user can expect to return 3% of all PAL devices purchased or three parts per hundred.

Predictable Programming

From a standpoint of device reliability, it is important for users to distinguish between the untestable behavior of fusible links and highly predictable and repeatable retention characteristics of the GAL device's E²CMOS arrays. Design features enable each E²CMOS cell to be checked for sufficient margin after programming, to ensure that cells comply with the retention requirements dictated by SGS-THOMSON quality and reliability goals. All devices undergo retention testing, in which every cell's charge is measured before and after a high-

temperature "bake", to detect any unusual degradation that might portend potential failures. For the user, this means that programmer "sensitives" are a thing of the past. In contrast, poorly processed fuse links in a bipolar PLD - which might require a higher current to program - can combine with a weak programmer (low current sourcing) to produce poor fusing characteristics. If proper fuse gapping is not achieved 100% of the time, reliability problems will arise. With fuse link technology, reliability can often be a function of programming hardware; with SGS-THOMSON E²CMOS cells, reliability is inherent. Once programmed, the GAL device is programmed for life (unless, of course, the user decides to erase and program it again).

Figure 24. Standard Product Manufacturing Flow



Testing GAL Devices

The GAL device standard manufacturing flow (figure 24) has been designed to test and stress 100% of the device circuitry over the full commercial or military temperature operating range. Only with E²CMOS technology is such an objective realizable, and SGS-THOMSON is one of few PLD manufacturers able to make that claim. To thoroughly check the GAL device, its circuits are partitioned into four sections, each of which is subjected to a specific test routine.

As shown in the GAL16V8 block diagram of figure 25, the four sections are: Input logic, Output Logic Macrocell (OLMC), E²CMOS cell array, and programming circuitry. The parameters and specifications of each of the four sections are thoroughly validated, as described in the following paragraphs.

Figure 25. GAL Device Block Diagram

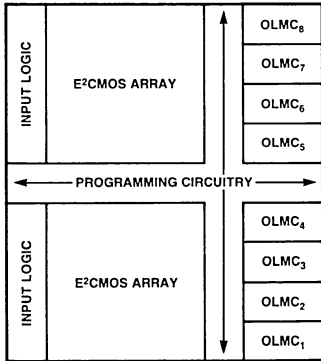
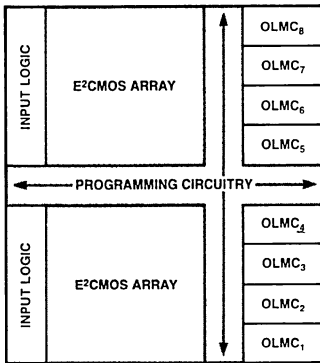


Figure 26. Basic Function Test Coverage



DC Parameters

All DC parameters are measured twice, with ample guardbands: during die-probing at the wafer level, and after the GAL devices are packaged. Tests include input leakage, I/O leakage, stand-by current, V_{IL} , V_{IH} , V_{OL} , I_{OL} , I_{OH} , and V_{OH} - all measured under worst-case bias conditions at both temperature (either commercial or military) and voltage limits.

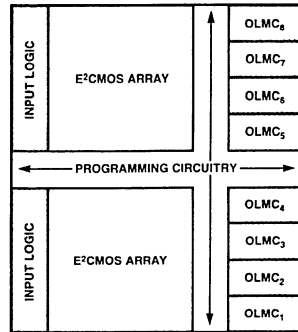
Basic Function

Basic functional testing verifies that the E²CMOS cell array and all programming circuitry is operational. Timings and levels associated with serial-shift register operation and array-addressing are used in conjunction with several data patterns to detect any array defects. During basic function testing the opportunity is also taken to test other GAL device features, such as register preload and power-on reset. The tinted region of figure 26 indicates the areas of basic function test coverage.

AC Performance

Since GAL device performance is taken very seriously, triple guardbands are used (voltage, timing and temperature) to ensure that device performance is well within specifications. Every timing specification is strictly tested, using worst-case patterns for each architectural configuration. By AC testing with different architectures, every AC path of the Output Logic Macrocell is tested. With AC performance testing, input logic and I/O feedback are verified, as is the AC integrity of the array. Figure 27 highlights the portions of the GAL16V8 tested in this step. Note that the patterns used for AC verification are placed into the normal array; no test rows or columns are used. In fact, so extensive is the testing performed on units prior to leaving the factory that SGS-THOMSON guarantees 100% AC yields.

Figure 27. AC Performances Test Coverage



Active Power

For the GAL device power measurement, operating current is maximized through use of a pattern that combines standby current and a peak transient power. The device is tested with an asynchronous

(16L8 type) architecture to take advantage of all available outputs. Eight inputs are cycled at a frequency in excess of 15MHz, while the supply voltage is maintained at its upper limit.

(Most applications will require as a little as half the power demanded by the above pattern).

Reliability

To optimize GAL device reliability, two special test features were incorporated into the GAL family: margin testing and internal verify. Margin testing provides the ability to individually measure the charge content of each cell, and is used primarily for two evaluations:

1) To measure each cell after programming, and thereby verify that sufficient voltage margin has been attained;

2) To measure cell charge before and after a high temperature stress, and thereby identify weak cells that would likely exhibit poor retention characteristics.

Internal verify provides the ability to monitor the voltage and current on internal nodes.

Stress-induced changes can be identified through internal verify, and a primary use of this feature is to monitor internal circuit nodes before and after a high voltage dynamic stress. Small changes in internal characteristics often forewarn premature circuit fatigue.

With the implementation of margin test and internal verify, reliability screens are made more effective. Used in conjunction with a dynamic and static stress, these two device features ensure dependable circuit operation and rock-solid E²CMOS cell retention.

Quality

To promote quality in the SGS-THOMSON GAL products, strict policies and procedures for manufacturing are closely adhered to. Electrostatic-discharge (ESD) levels are continuously monitored on samples taken at each manufacturing step.

100% actual step test, using generous guardbands in temperature, timings, signal levels and supply voltage, guarantees that all shipped material meets published specifications.

Testability, Quality and reliability were foremost objectives during the design, development and manufacturing of the GAL family.

The time and energy expended on this product indeed support that 100% programming and functional yield are hard-earned facts-not just claims.

POWER CONSIDERATION

Beginning with an understanding of some fundamental power-consumption characteristics of bipolar and CMOS circuit technologies, a technique can be developed that allows system designers to accurately predict typical and worst-case power consumption of GAL devices in a specific application. Here, we investigate the components of GAL device power consumption - and its dependence on the programming pattern - to develop such a structured technique.

Power consumption in integrated circuits comprises of two fundamental components: A static (DC) component and a dynamic (AC) component. The total power consumed by an integrated circuit in a system is simply the sum of these two components:

$$P_D = P_{DAC} + P_{DDC}$$

$$= (V \cdot I_{AC}) + (V \cdot I_{DC})$$

$$= V (I_{AC} + I_{DC})$$

where: P_D is the total power dissipation, I_{AC} is the dynamic (switching) current consumption, I_{DC} is the static (standby) current consumption, and V is the operating supply voltage.

Dividing both sides by the operating voltage (V) yields the total current:

$$I_{CC(TOTAL)} = I_{AC} + I_{DC}$$

The I_{AC} term can be further simplified:

$$I_{AC} = \frac{dQ_{AC}}{dt} = C_{TOTAL} \frac{dV}{dt} = (C_L + C_{EQ}) \frac{dV}{dt}$$

where C_L is the total output load capacitance being driven by the integrated circuit, C_{EQ} is the total equivalent capacitance presented to the system by the integrated circuit due to the internal device nodes being charged and discharged during switching, dV is equal to the supply voltage, and dt = 1/f (where f = switching frequency in Hz). Therefore:

$$I_{AC} = (C_L + C_{EQ})Vf$$

$$I_{CC(TOTAL)} = (C_L + C_{EQ}) Vf + I_{DC}$$

It can be seen that I_{DC} is a constant, while I_{AC} is linear with respect to frequency, voltage, and load capacitance. In order to simplify the following discussion. We can assume an unloaded integrated circuit and set C_L = 0. We are now left with the equivalent capacitance (C_{EQ}) presented to the system by the integrated circuit:

$$I_{CC(TOTAL)} = C_{EQ} Vf + I_{DC}$$

Bipolar integrated circuits have historically been dominated by the I_{DC} term and have, therefore, ignored the I_{AC} term (commonly referred to as the 'CVf' term.) With the advent of bipolar technologies capable of attaining operating frequencies of 50-100 MHz, this "CVf" term is no longer negligible. Of course, when considering high-performance CMOS technologies, this CVf term is often the dominant component of total I_{CC} . In fact, many CMOS devices now separate these two components out and supply separate specifications:

$$I_{SB} = I_{CC} \text{ standby (or } I_{DC} \text{ specified in mA)}$$

$$I_{AC} = \text{"CVf" component of } I_{CC} \text{ (in mA/MHz)}$$

The intent is to allow a designer to determine the total I_{CC} of a specific IC in a specific application. Programmable Logic Devices (both bipolar and CMOS) have complicated matters for the users and manufacturers of these devices, owing to their inherent configurability. Each unique PLD pattern may generate significantly different values for both I_{SB} and CVf power, depending on the functions and features utilized in each application. The SGS-THOMSON GAL device family has been thoroughly characterized in terms of both of these components, in order to provide a simple and accurate technique for estimating total I_{CC} in a specific application.

Separating I_{SB} and I_{AC} Components

Each GAL device can be divided into two distinct areas of circuitry: one that consumes DC power and one that doesn't. The high-speed single ended sense amplifier section consumes DC power (I_{SB}) to attain performance through its ability to sense and amplify internal signal swings of as little as 50mV to full-supply-level logic signals. (Additionally, the oscillator circuitry that generates the negative substrate bias contributes about 2 mA to I_{SB} .) All remaining circuitry is 100% CMOS technology, which essentially generates no I_{SB} component. This circuitry includes input buffers, row drivers, output logic macrocells, and output drivers. While contributing negligible I_{SB} , these circuits have direct and measurable contributions to C_{EQ} Vf current.

Standby current (I_{SB})

E²CMOS GAL devices employ a high-speed sense amplifier on each product term in the array. Each sense amplifier typically consumes ~560µA of standby bias current. On a GAL 16V8 there are sixty-four product terms, such that a typical I_{SB} contribution is:

$$(64 \text{ PT}) (0.56 \text{ mA/PT}) = 36\text{mA}$$

Adding the 2mA for the oscillator circuit yields:

$$\text{Total } I_{SB} = 36\text{mA} + 2\text{mA} = 38\text{mA}$$

Due to variations in wafer fabrication process parameters, this "typical" number will vary approximately ±20%. This variation of typical I_{SB} versus process parametric spread at 5.25V and 25°C is illustrated in Figure 28 for both half-power GAL devices.

C_{EQ} Vf Current (I_{AC})

Due to the use of E²CMOS technology, each of the many different paths and functions available to a system designer contributes to the C_{EQ} Vf current only when actively switching.

To accurately estimate this C_{EQ} Vf current in a GAL device requires partitioning it into independent fundamental components that can be treated separately, then added together for a net contribution. There are basically three distinct circuit areas that contribute to C_{EQ} Vf current:

- 1) Switching input and I/O buffers (includes rows in the array)
- 2) Switching product terms (Columns in the array)
- 3) Switching output logic macrocells and output drivers

Each of these circuit blocks represents a significant capacitance that must be charged or discharged each time it switches logic states. The C_{EQ} Vf current may be expressed as the sum of these components:

$$C_{EQ}Vf = (C_1 + C_2 + C_3) Vf$$

$$= (C_1V + C_2V + C_3V)f$$

Where C_1 , C_2 , C_3 refer to equivalent capacitances for the above-referenced components.

Each of the three CVf components has been characterized as a function of frequency and provided for reference in Figure 29,30 and 31. Figure 29 provides I_{AC} characteristics as a function of the number of switching device input or I/O pins. Figure 30 illustrates the I_{AC} characteristics as a function of the number of switching product terms. Finally, Figure 31 provides I_{AC} characteristics as a function of the number of switching outputs. Table 2 provides a summary of these components.

Table 2. Summary of AC Current Components

Term	Half Power GAL I_{AC} Contribution	Quarter Power GAL I_{AC} Contribution	Description
C_1V	30µA/MHz 40µA/MHz	30µA/MHz 40µA/MHz	Per switching input Per switching I/O
C_2V	7µA/MHz	6µA/MHz	Per switching product term
C_3V	100µA/MHz	50µA/MHz	Per switching output

Figure 28. Standby Current vs Process Parameters

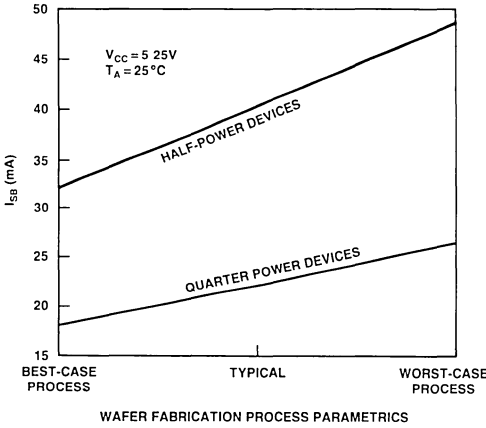


Figure 29. AC Current vs. Inputs Switching

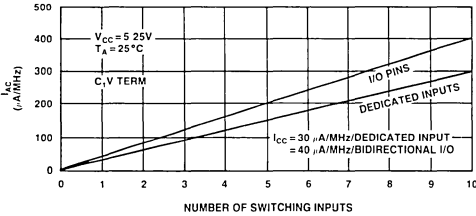


Figure 30. AC Current vs. Product terms Switching

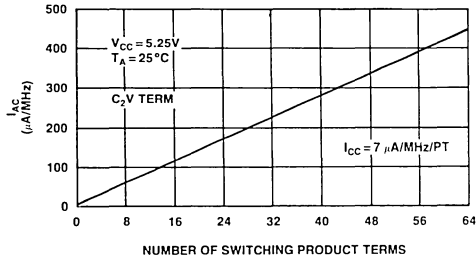
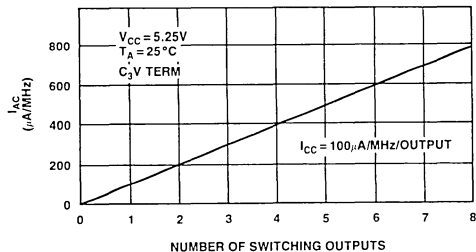


Figure 31. AC Current vs. Outputs Switching



I_{CC}(TOTAL)

It now becomes a simple task to calculate total I_{CC} for a specific application by merely summing the components, according to the following equation:

$$I_{CC}(TOTAL) = I_{SB} + (C_1VI + C_2VP + C_3VO)f$$

where I = # of switching inputs
 P = # of switching product terms
 O = # of switching outputs

To try a specific example, consider a typical 10MHz application of a half-power GAL 16V8. A system application may utilize eight inputs, four outputs, two I/Os, with sixteen product terms utilized. Substituting into the equation yields:

$$\begin{aligned} I_{CC}(TOTAL) &= 38mA + [(30\mu A/MHz)(8) + (7\mu A/MHz)(16) + (100\mu A/MHz)(4) + (40\mu A/MHz)(2)]10 \text{ MHz} \\ &= 38mA + (240\mu A/MHz + 112\mu A/MHz + 400\mu A/MHz + 80\mu A/MHz) 10MHz \\ &= 38mA + (832\mu A/MHz) 10MHz \\ &= 38mA + 8.32mA \\ &= 46.32mA \end{aligned}$$

This is significantly less than the data sheet specification of 90mA. There are three considerations which explain this discrepancy:

- 1) This application is a 10MHz application. The data sheet specifies I_{CC} at 15MHz.
- 2) This application does not use all inputs and I/Os, all product terms, and all outputs. The “CV” component of I_{CC} is, therefore, significantly reduced.
- 3) The effects of temperature have not yet been included. Figure 32 provides normalized I_{CC} as a function of ambient temperature (normalized to 25°C). To calculate the worst-case power for the example above at -55°C requires multiplying by the temperature coefficient Θ_T:

$$\begin{aligned} I_{CC}(TOTAL, \text{WORST CASE}) &= I_{CC}(\text{TYPICAL}) \cdot \Theta_T \\ &= (46.32mA) (1.19) \\ &= 55.1mA \end{aligned}$$

Now, for illustration, look at the worst-case I_{CC} possible on a half-power GAL device. This requires:

- T_A = -55°C
- # of switching inputs = 8
- # of switching I/O = 8
- # of switching PTs = 64
- # of switching outputs = 8
- System frequency = 15MHz

Plugging into the equation yields:

$$I_{CC}(TOTAL) = (48mA + [(30\mu A/MHz)(8) + (7\mu A/MHz)(64) + (100\mu A/MHz)(8) + (40\mu A/MHz)(8)]15MHz)1.19$$

$$= (48\text{mA} + [240\mu\text{A}/\text{MHz} + 448\mu\text{A}/\text{MHz} + 800\mu\text{A}/\text{MHz} + 320\mu\text{A}/\text{MHz}] 15\text{MHz}) 1.19$$

$$= (48\text{mA} + 27.1\text{mA}) 1.19$$

$$= (75.1\text{mA})(1.19) = 89.37\text{mA}$$

This is quite close to the maximum I_{CC} of 90mA quoted on the data sheet.

It is now a simple task to generate a chart that graphically compares the total operating I_{CC} of GAL devices with that of bipolar PLDs. The following scenarios are illustrated in Figure 33:

- A) Bipolar "A-PAL" operating I_{CC}
- B) Half-power GAL 16V8 with a "typical" pattern
- C) Half-power GAL 16V8 with a "worst case" pattern
- D) Quarter-power GAL 16V8 with a "typical" pattern.

Only room-temperature curves are illustrated, as both bipolar PLDs and E²CMOS GAL devices have similar temperature coefficients for I_{CC} . It can be seen that the half-power GAL16V8 significantly outperforms the bipolar PLD in terms of operating I_{CC} (by typically 90mA per device during standby, and by 75mA per device at 30MHz).

Figure 32. Normalized Current vs. Temperature

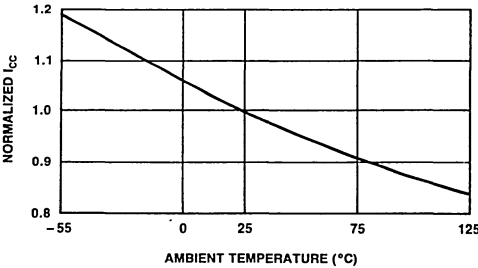
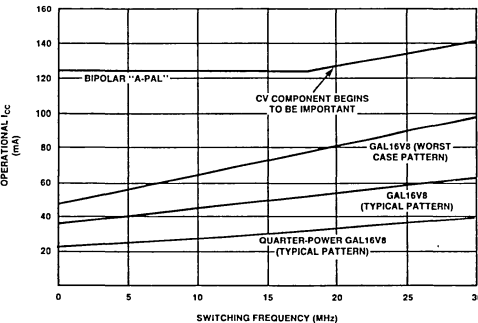


Figure 33. PLD Current vs. Frequency



LATCH-UP PROTECTION

Latch-up is destructive bipolar-device action that can potentially occur in any CMOS-processed device. It is characterized by extreme runaway supply current and consequential smoking plastic packages. Latch-up is peculiar to CMOS technology, which integrates both P and N channel transistors on one chip.

In the doping profile of a CMOS inverter, parasitic bipolar (PNPN) silicon-controlled-rectifier (SCR) structures are formed. Figure 34 shows the process cross section of a CMOS inverter, as well as the bipolar components to the parasitic SCR structure. In steady-state conditions, the SCR structure remains off. Destruction results when stray current injects into the base of either Q₁ or Q₂ in Fig. 34.

The current is amplified with regenerative feedback (assuming that the beta product of Q₁ and Q₂ is greater than unity), driving both Q₁ and Q₂ into saturation and effectively turning on the SCR structure between the device supply and ground. With the parasitic SCR on, the CMOS inverter quickly becomes a nonrecoverable short circuit; metal trace lines melt and the device becomes permanently damaged.

Causes of Latch-Up

It has been explained that parasitic bipolar SCR structures are inherent in CMOS processing. If triggered, the SCR forms a very low-impedance path from the device supply to the substrate, resulting in the destructive event. Two conditions are necessary for the SCR to turn on: The beta product of Q₁ and Q₂ must be greater than unity, which, although minimized, is usually the case; and a trigger current must be present.

The cause of latch-up is best understood by examining the mechanisms that produce the initial injection current to trigger the SCR network. Figure 35 is a schematic of the parasitic bipolar network present in a CMOS inverter, where node "b" is the inverter output. It can be seen that two events might trigger latch-up: 1) The inverter output could overshoot the device supply, thereby turning on Q₃ and injecting current directly into the base of Q₂; and 2) the inverter output could undershoot the device ground, turning on Q₂ immediately. However, a third condition could also trigger latch-up: if the supply voltage to the P + diffusion were to rise more quickly than the N-well bias, Q₁ could turn on. Within the device circuitry, overshoot and undershoot can be controlled by design. A problem area exists at the device inputs, outputs and I/Os because external conditions are not always perfect. Powering up can also be a potential problem because of unknown bias conditions that may arise.

With CMOS processing the possibility of latch-up is always present. The major causes of latch-up are understood and it is clear that if CMOS is to be used, solutions to latch-up will have to be created. As the technology evolves, solutions to latch-up are becoming more creative.

Figure 34. CMOS Inverter Cross-Section

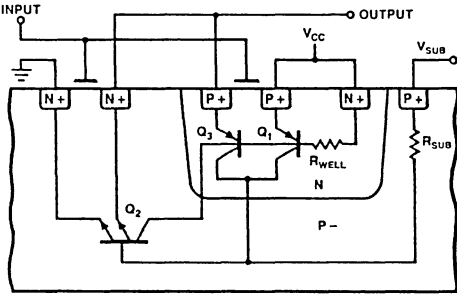
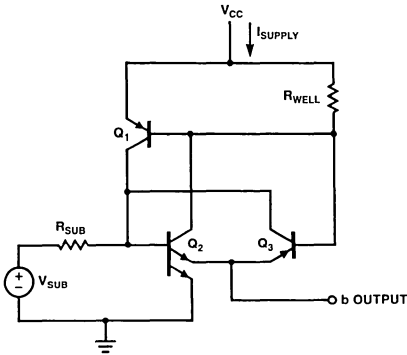


Figure 35. Parasitic SCR Schematic



Approach to Eliminate Latch-Up

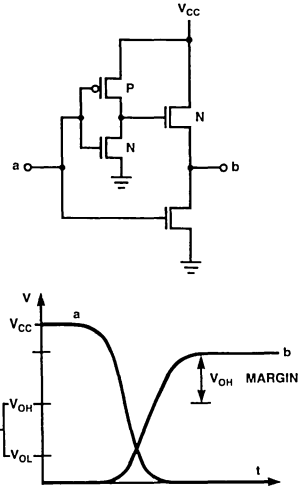
The intent of the GAL family was to implement cost-effective solutions to each major cause of latch-up. The goal was met through three device features.

The most susceptible areas for latch-up are the device inputs, outputs and I/Os. Extreme externally applied voltages may cause a P + N junction to forward-bias, leading to latch-up.

To prevent latch-up by large positive swings on the device inputs, ESD protections don't use P + diffusion diode connected to VCC to safe device outputs or I/O pins, NMOS output drivers were used. This eliminates the possibility of turning on Q3 (Figure 35) with an output bias in excess of the device supply voltage. Figure 36 contains the effective NMOS output driver and its switching characteristics. Note that the output does not fully reach the supply voltage, but still provides adequate VOH margin for TTL compatibility.

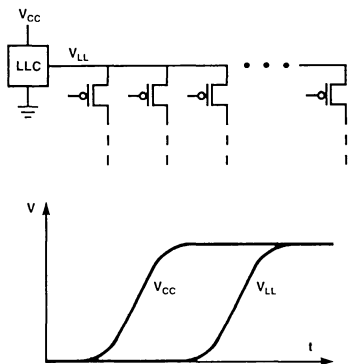
To prevent negative swings on device input, output and I/O pins from forward-biasing the base-emitter junction of Q2, a substrate-bias generator was employed. By producing a VSUB of approximately -2.5V, undershoot margin is increased to about -3V.

Figure 36. NMOS Output Driver



To insure that non undesired bias conditions occur with P + diffusions, SGS-THOMSON has developed the power-up circuitry, illustrated in Figure 37. In short, the drain of all P channel devices normally connected to the device supply, are now connected to an alternate supply that powers up after the device N-wells have been biased and the substrate has reached its negative clamp value. This prevents any hazardous bias conditions from developing in the power-up sequence. After power-up is complete, the power-up circuitry becomes dormant until a full power-down has occurred; this eliminates the chance of an unwanted P channel power-down during device operation.

Figure 37. Power-Up Circuitry



To determine the amount of latch-up immunity achieved with the three device features utilized in the GAL family, an intensive investigation was carried out. Each step was conducted at 25° and 100°C; inputs, outputs, and I/Os were sequentially forced to -8V and +12V while the device underwent fast and slow power-ups; devices were repeatedly “hot-socket” switched with up to 7.0V.

Even under the extreme conditions specified, no instance of latch-up, occurred. In attempt to provoke latch-up ±50mA was forced into each output and I/O pin. The device output drivers were damaged in the battle, and still latch-up was not induced.

Based on the data, it is evident that the GAL family is completely immune to latch-up, even when subjected to a wide variety of extreme conditions, including current at inputs, outputs, and I/Os, power-supply rise time, hot-socket power-up and temperature.

QUALITY AND RELIABILITY

QUALITY PROGRAM

The quality and reliability of a product depend on a range of activities starting from the conception and design of a new product through to the production shipment and the follow-up given to the customer.

It is well known that reliability must be designed into the product and the process.

But to manufacture consistently reliable high quality products, SGS-THOMSON believes that it is essential for everyone in the company to appreciate the importance of maintaining and improving the levels of Quality and Reliability.

SGS-THOMSON has adopted a "Total Quality Control" approach which means that different departments must work together with the aim of improving Quality. With this approach potential problems can be solved at the stage where they arise, so that defects are not passed on to the finished product.

Total Quality Control assures the prevention of quality problems rather than simply eliminating defective finished products.

The customer benefits from this approach because it guarantees better quality and reliability and reduces costs.

Reliability

Semiconductor devices must function normally in a stable manner throughout the specified life of the product.

SGS-THOMSON therefore takes meticulous care in the design and manufacturing stages and analyzes the different factors that effect the Reliability of semiconductors.

Component reliability is described in quantitative terms by measuring the failure rate as a function of time.

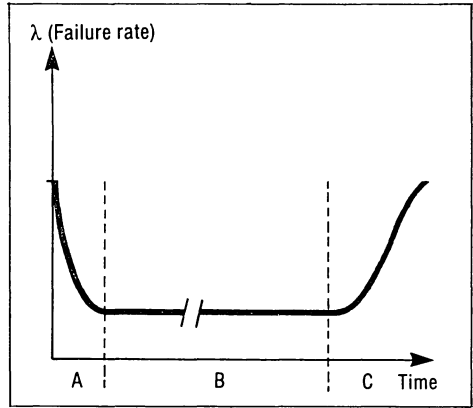
The failure rate distribution of a typical device follows the familiar bathtub curve shown in Fig. 1. This curve is divided into three zones.

Zone A covers the so called "infant mortality" period. The predominant failure mechanism are related to assembly defects. Actions and checks throughout the process allow SGS-THOMSON to reduce infant mortality failure.

Zone B represents the random failure portion of the curve related to the useful life of the device. This duration, generally very long, depends on the applied stresses such as temperature, voltage, and power, etc.

In Zone C there are wearout mechanism resulting in parametric and catastrophic failures.

Fig. 1 - Failure rate distribution curve



Reliability testing

There are two types of reliability tests: those performed during design and development and those carried out in production. The first type is usually performed on a small sample but for long periods of time under highly accelerated conditions in order to investigate wearout failures and to determine tolerances and limits of the design. The second type of tests are performed periodically during production to check, maintain and improve the assured Quality and Reliability levels. All reliability tests performed by SGS-THOMSON are under more severe conditions than those met in the field. These conditions although accelerated, are chosen to simulate stresses that devices are subjected to in actual operation, and care is taken to ensure that the failure modes are unchanged.

Accelerated tests

Through accelerated stress it is possible to evaluate the failure rates of the components, as a percentage of devices expected to fail every 1000 hours under test conditions. Derating the data at different conditions, we can estimate the life expectancy under actual operating conditions. In its simplest form the failure rate (F.R.) under given conditions is:

$$F.R. = \frac{N}{DH} \text{ where } \begin{matrix} N = \text{No of failures} \\ D = \text{No of devices} \\ H = \text{No of testing hours} \end{matrix}$$

If we intend to determine the failure rate under different conditions an acceleration factor must be considered. Data obtained for device families under accelerated conditions give a relationship between the stress applied and failure rate that is supported by data collected in the field. Data collected from operating life tests performed at high temperature are derated for normal operating conditions and the failure rate may be estimated with sufficient accuracy.

Activation energies are in the range 0.3 to 0.6 eV (typically 0.44 eV) for infant mortality failures and 0.6 to 1.4 eV (typically 1 eV) for random failures.

PPM (Results and targets)

As a consequence of its quality improvement programmes SGS-THOMSON has continually improved outgoing quality and is pursuing ambitious quality targets.

PPM values and targets for cumulative electrical failures* (inoperative mechanical included)

YEAR	1987	1988	1989	1990
PPM	75	40	25	10 ÷ 20

* Values referred to the end of years.

Fig. 2 - Life-Hours

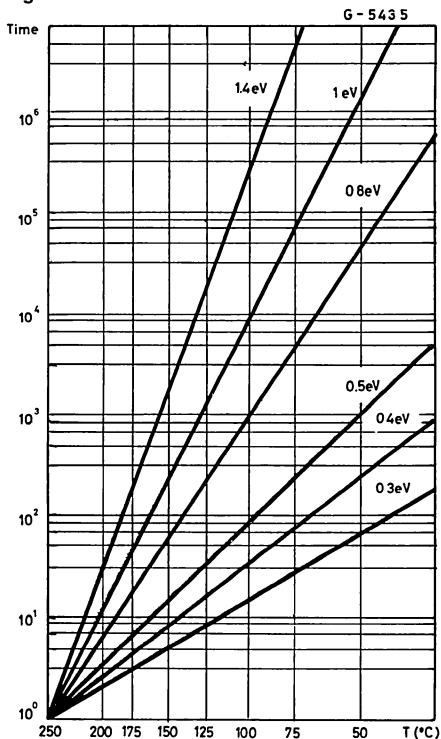


Figure 2 shows temperature derating curves and the multiplying factors for temperature reduction based on the assumption of the Arrhenius law:

$$F.R. = A e^{-E_a/KT_j} \tag{1}$$

- A = Constant
- K = Boltzman's constant
- Ea = Activation energy
- tj = Absolute temperature

For two different temp. $F.R.(T_1) = F(T_1, T_2) F.R.(T_2)$

from (1) it is: $F(T_1, T_2) = \text{EXP} \left[\frac{-E_a}{K} \left(\frac{1}{T_1} - \frac{1}{T_2} \right) \right]$

Field Programmable Logic Devices (FPLD)

Subgroup	Parameters	Insp. level	AQL
A1	Visual and mechanical inspection	I	0.04*
A2 + A3 + A4	Cumulative electrical and inoperative mechanical failures	II	0.04

* - 0.065 for SMD (Surface Mounted Devices)

- Guaranteed temperatur range: according to SGS-THOMSON data books.

Product qualification

A “no compromise” strategy was adopted at each stage of production from development to preproduction, to full production stage; quality and reliability were always priority targets.

We have envisaged 3 main check points:

1. project validation at design stage
2. prototypes qualification
3. final product qualification.

At design stage CAD tools (for simulation and design rule checks) and quality council meeting are used for continuous monitoring of quality aspects of the circuits.

Prototypes qualification is performed by means of an agreed and fixed set of tests for reliability and product characterization.

The final outputs are:

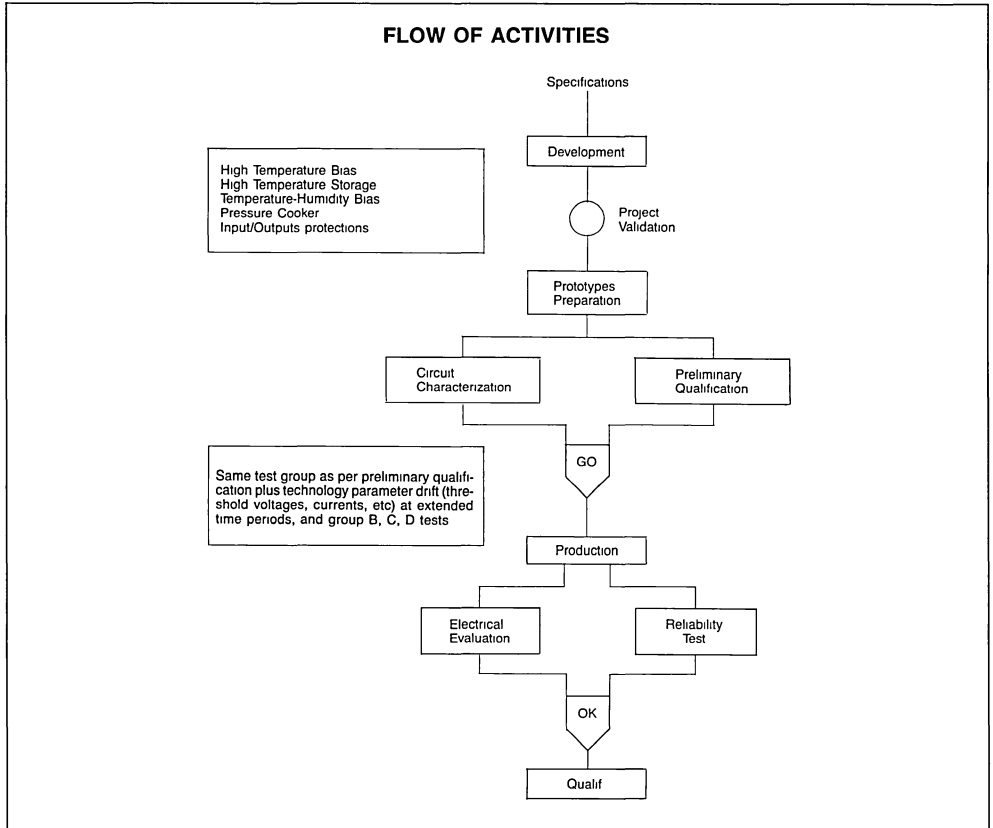
- a) circuit characterization manual
- b) circuit qualification report
- c) failure analysis reports (when needed).

See below for operative data and flow.

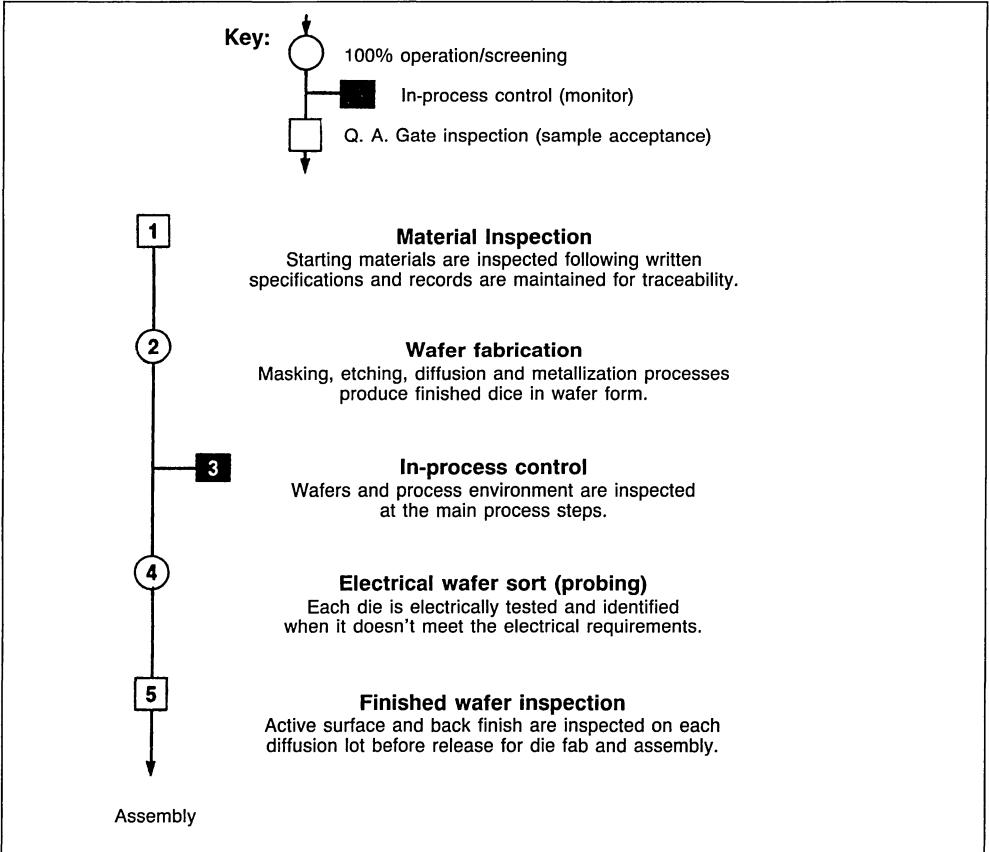
Finally product qualification at production stage is obtained with a set of reliability tests mainly based upon:

- d) accelerated temperature life tests
- e) high temperature storage
- f) temperature-humidity accelerated conditions
- g) high voltage stresses (inputs-outputs protection, latch-up).

Both circuit and process parameters are monitored.



WAFER FAB STANDARD PRODUCTION PROCESS FLOW CHART

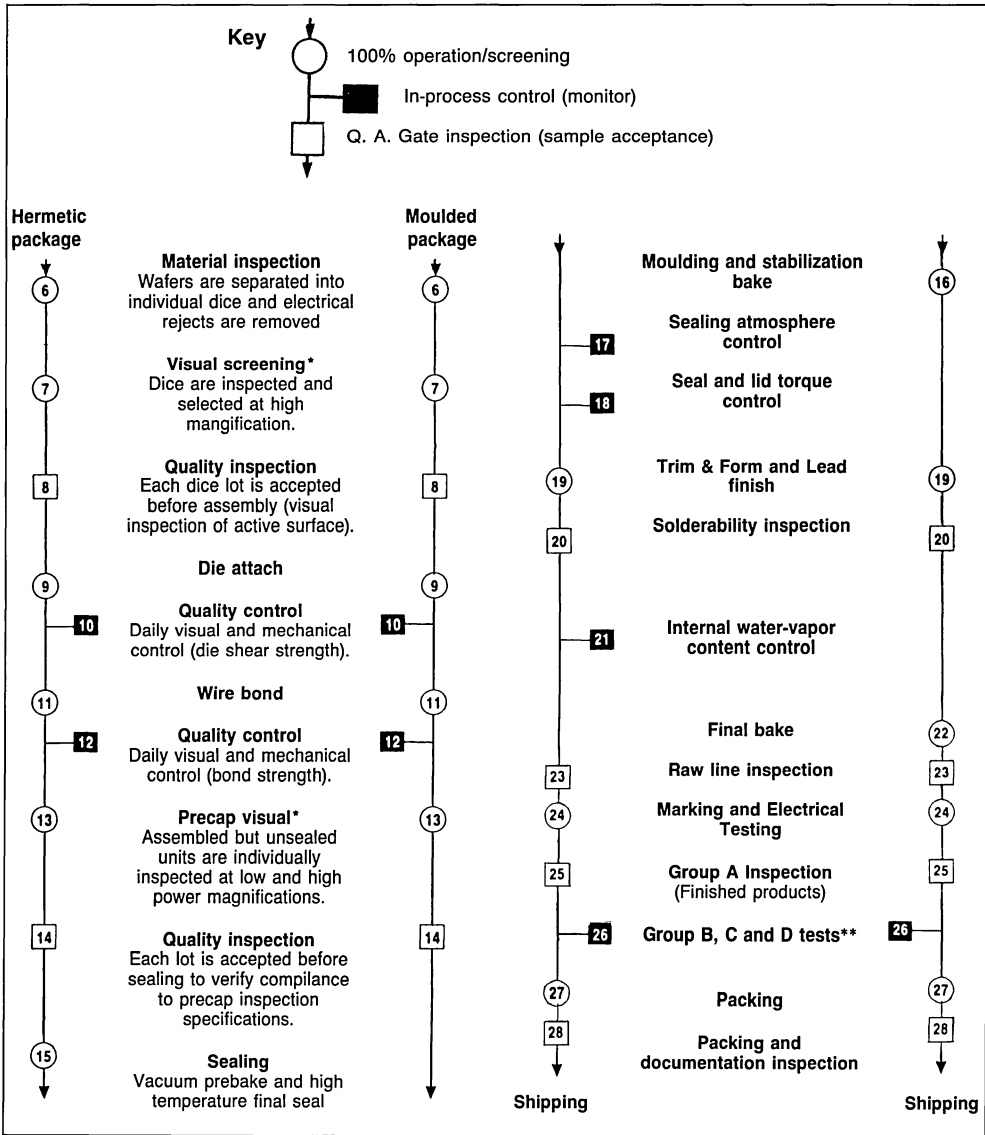


QUALITY INSPECTIONS/MONITORS DURING WAFER FABRICATION

The Table emphasizes the most important fabrication steps with the relevant Quality inspections/monitors performed.

PROCESS STEPS	IN-PROCESS INSPECTIONS/MONITORS
OXIDATION	<ul style="list-style-type: none"> — Visual — Thickness — CV plot (Stability of ionic concentration and contamination control)
DEPOSITION: Nitride, BPSG, PSG, Poly Si	<ul style="list-style-type: none"> — Visual — Thickness — Doping content
PHOTO LITHOGRAPHY	<ul style="list-style-type: none"> — Mask and wafer cleanliness — Visual (alignment and focusing accuracy) — Critical dimensions
ETCHING	<ul style="list-style-type: none"> — Visual (quality of etching and wafer cleanliness) — Critical dimensions
DOPING BY IMPLANT (P, As, B)	<ul style="list-style-type: none"> — Sheet resistance (dose and implant uniformity)
DOPING BY DIFFUSION (POCl ₃)	<ul style="list-style-type: none"> — Sheet resistance — Thickness — CV plot (Stability of ionic concentration and contamination control)
METALLIZATION	<ul style="list-style-type: none"> — Visual — SEM (step coverage and film quality) — Thickness — CV plot (stability of ionic concentration and contamination control)
PASSIVATION	<ul style="list-style-type: none"> — Thickness — Phosphorus concentration — Passivation integrity (density of pinholes and cracks) — Visual
BACK FINISHING	<ul style="list-style-type: none"> — Wafer thickness
ELECTRICAL CHARACTERIZATION	<ul style="list-style-type: none"> — Threshold voltage — Electrical Characteristics: e.g.: leakage current, breakdown voltage, resistivity, etc.
WAFER INSPECTION	<ul style="list-style-type: none"> — Visual (microscope and/or laser surface inspection system)

ASSEMBLY STANDARD PRODUCTION PROCESS FLOW CHART



* Omitted when the intrinsic quality meets the specified quality level.

** For non military products, these reliability tests can be performed after step 23 on 100% electrically tested samples (when requested).

QUALITY TESTS DURING ASSEMBLY PROCESS

Process steps	Tests	Descriptions
10	QUALITY CONTROL (die attach)	— MIL STD 883C Method 2010 cond. B (internal visual) and Method 2019 (die shear strength)
12	QUALITY CONTROL (bonding)	— MIL STD 883C Method 2010 cond. B (internal visual) and Method 2011 cond. D (bond strength)
14	QUALITY INSPECTION (precap)	— MIL STD 883C Method 2010 cond. B (internal visual)
17	SEALING ATMOSPHERE CONTROL	Moisture content: < 200 ppm for Ceramic packages
18	SEAL CONTROL	Fine Leak: — MIL STD 883C Method 1014 cond. A1 Helium leak detector after pressurization in He for 2 h at 4.2 atm Limit: $5 \cdot 10^{-8}$ cc/s for ICV* < 0.4 cc $2 \cdot 10^{-7}$ cc/s for ICV \geq 0.4 cc * ICV = internal cavity volume Gross Leak — MIL STD 883C Method 1014 cond. C1 (fluorocarbon gross leak) 5 Torr vacuum for 1h followed by pressurization of the devices immersed in mineral oil at 4.2 atm for 2h, and subsequent immersion in mineral oil at Ta = 125°C
	LID TORQUE CONTROL	Ceramic packages only MIL STD 883C Method 2024 (e.g. \geq 60 kg · cm for seal area values between 1.41 and 1.73 cm ²)
20	SOLDERABILITY INSPECTION	— MIL STD 883C Method 2003 Soldering temperature $245 \pm 5^\circ\text{C}$ for 5 ± 0.5 sec. with preconditioning of 1h (1) above boiling distilled water and 5 to 10 sec. in rosin base flux
21	INTERNAL WATER VAPOR CONTENT CONTROL	Dew Point method MIL STD 883C Method 1018 procedure 3 5000 ppm max (dew point temperature less than -15°C) Ceramic packages only.
22	FINAL BAKE	For SMD only (according to internal specifications)
23	RAW LINE INSPECTION	External Visual MIL STD 883C Method 2009 Note: at this step some reliability tests (pressure pot, temperature cycling, life test etc.) are performed as a monitor, generally on a weekly basis, to have fast feedback on process behaviour (Real Time Control Tests)

QUALITY TESTS DURING ASSEMBLY PROCESS (Continued)

Process steps	Tests	Descriptions
25	GROUP A INSPECTION	— See page 200
26	GROUPS B, C AND D TESTS	Performed on the product family representative types (by rotation); the results are extended to all the other devices of the same family according to the structure similarity concept
28	PACKING AND DOCUMENTATION INSPECTION	Inspection for: — right quantity — right type — right boxing — right labelling — right documentation — various

(1) This preconditioning is going to be substituted by 8h as per MIL-STD-883 C requirements
 Please consult our sales organization for any further information.

GROUPS B, C AND D RELIABILITY TESTS

Every week or every 3 months on raw line material and/or finished products

Test	MIL-STD-883 C		LTPD
	Method	Condition	
SUBGROUP 1 physical dimensions	2016	Major dimensions according to data sheet	2 devices (no failure)
SUBGROUP 2 (1) resistance to solvents	2015	1 minute immersion in solvent solution followed by 10 strokes with a soft brush (the procedure shall be repeated 3 times) solvent solution 2.1a only for moulded packages.	4 devices (no failure)
SUBGROUP 3 (1) solderability (*)	2003	Soldering temperature $245 \pm 5^\circ\text{C}$ for 5 ± 0.5 sec. with preconditioning of 1 h above boiling distilled water and 5 to 10 sec. in rosin base flux	10
SUBGROUP 4 high temperature operating life test high temperature retention high VT high temperature retention low VT extended endurance	1005	1000 h at $T_a = 125^\circ\text{C}$, $V_{CC} = 5,5 \text{ V}$ 1000 h at $T_a = 150^\circ\text{C}$ $V_{CC} = 5.5\text{V}$ 1000 h at $T_a = 150^\circ\text{C}$ $V_{CC} = 5.5\text{V}$ 1000 cycles $T_a = 25^\circ\text{C}$ $V_{CC} 5.5\text{V}$	5**

(1) Performed weekly on finished product
 (*) See note 1 page 206
 (**) Different samples for each test.

GROUPS B, C AND D RELIABILITY TESTS (Continued)

Every week or every 3 months on raw line material and/or finished products

Test	MIL-STD-883 C		LTPD
	Method	Condition	
SUBGROUP 5 (hermetic packages only) temperature cycling constant acceleration seal - fine - gross end-point electrical parameters	1010 2001 1014	test condition C (10 cycles $T_a = -65^{\circ}\text{C}$ to $+150^{\circ}\text{C}$); 10 minutes at temperature extremes; recovery time max 15 minutes after 1 minute max transfer time test condition E (30000 g) Y1 orientation only (2) test condition A1 (see step 18 pag. 205) test condition C1 (see step 18 pag. 205) as specified in the applicable device specification	10
SUBGROUP 6 (1) (moulded packages only) pressure pot end-point electrical parameters		$T_a = 121^{\circ}\text{C}$, 2 atm, 168 h minimum as specified in the applicable device specification	10
SUBGROUP 7 (1) (moulded packages only) HAST (Highly Accelerated Stress Test)	—	130°C -85% RH with bias $t = 48$ h according to detail specification as specified in the applicable device specification	10
SUBGROUP 8 (hermetic packages only) seal - fine - gross thermal shock seal - fine - gross end-point electrical parameters	1014 1011 1014	test condition A1 (see step 18 pag. 205) test condition C1 (see step 18 pag. 205) test condition B, 60 cycles ($T_a = -55^{\circ}\text{C}$ to $+125^{\circ}\text{C}$) 5 minutes at temperature extremes transfer time ≤ 10 sec. test condition A1 (see step 18 pag. 205) test condition C1 (see step 18 pag. 205) as specified in the applicable device specification	10

(1) Performed weekly on finished product

(2) 20000 g for packages with cavity perimeter of 5 cm or more and for with a mass of 5 grams or more

GROUPS B, C AND D RELIABILITY TESTS

Every 6 months on raw line material and/or finished products

Test	MIL-STD-883 C		LTPD
	Method	Condition	
SUBGROUP 1 lead integrity seal (hermetic packages only) - fine - gross	2004	test condition B2 (lead fatigue) - dual-in-line moulded packages; three leads shall be bent, 3 times, simultaneously for at least 15° permanent bend, returning then to the original position	10
	1014	test condition A1 (see step 18 pag. 205) test condition C1	
SUBGROUP 2 (hermetic packages only) thermal shock temperature cycling moisture resistance seal - fine - gross visual examination end-point electrical parameters	1011	test condition B; 15 cycles (Ta = -55°C to +125°C) 5 minutes at temperature extremes transfer time ≤ 10 sec.	10
	1010	test condition C; 100 cycles (Ta = -65°C to +150°C) 10 minutes at temperature extremes; recovery time max 15 minutes after 1 minute max transfer time	
	1004	Lead bend stress initial conditioning followed by 10 cycles of 24h; Ta = 25°C to 65°C RH = 80% to 100% one 3h cycle at Ta = -10°C	
	1014	test condition A1 (see step 18 pag. 205) test condition C1	
		per visual criteria of method 1004 and 1010 as specified in the applicable device specification	
SUBGROUP 3 (hermetic packages only) mechanical shock vibration, variable frequency constant acceleration (1) seal - fine - gross visual examination end-point electrical parameters	2002	test condition B; 1500 g - 0.5 msec. - 5 blows in each of the 6 orientations - not operating	10
	2007	test condition A; 20 g - 3 orientations f = 20 to 2000 cps; four 4 minutes cycles; 48 minutes total - not operating	
	2001 1014	test condition E (30000 g), Y1 orientation only test condition A1 (see step 18 pag. 205) test condition C1	
		per visual criteria of Method 1011 or 1010 as specified in the applicable device specification	

GROUPS B, C AND D RELIABILITY TESTS

Every 6 months on raw line material and/or finished products (continued)

Test	MIL-STD-883 C		LTPD
	Method	Condition	
SUBGROUP 4 salt atmosphere	1009	test condition A; 10 to 50 gr of NaCl per square meter per day for 24 h at Ta = 35°C	
seal (hermetic packages only) - fine - gross	1014	test condition A1 test condition C1 (see step 18 pag. 205)	
visual examination		per visual criteria of method 1009	
SUBGROUP 5 (moulded packages only)			
temperature cycling	1010	test condition C; 100 cycles (Ta = -65 to + 150°C 10 minutes at temperature extremes; recovery time max 15 minutes after 1 minute max transfer time	5
visual examination		per visual criteria of Method 1010	
end-point electrical parameters		as specified in the applicable device specification	
SUBGROUP 6 (moulded packages only)			
humidity test	CECC 90000	85°C/85% RH with bias, t = 1000 h according to detail specification	5
end-point electrical parameters		as specified in the applicable device specification; measurements at 0, 168, 500 and 1000 h	
SUBGROUP 7 (hermetic packages only)			
internal water-vapor content	1018	dew point method-procedure 3 (5000 ppm max)	3 devices 0 failures or 5 devices 1 failure (2)
SUBGROUP 8 (ceramic packages only)			
lid torque (3)	2024	(see step 18 pag. 205)	10

1) 20000 g for packages with cavity perimeter of 5 cm or more and/or mass of 5 grams or more.

2) Test three devices if one fails test two additional devices with no failure.

3) lid torque test shall apply only to packages which use a glass-frit seal to lead frame, lead or package body (i.e. wherever frit seal establishes hermeticity or package integrity)

QUALITY AND RELIABILITY

GROUPS B, C AND D RELIABILITY TESTS

Additional tests done during qualification:

Test	MIL-STD-883 C	
	Method	Condition
Electrostatic discharge sensitivity	3015	C = 100 pF R = 1.5K Ω
Latch susceptibility		V = ± 10 per pin

SALES OFFICES

AUSTRALIA

NSW 2027 EDGECLIFF
Suite 211, Edgecliff centre
203-233, New South Head Road
Tel (61-2) 327 39 22
Telex 071 126911 TCAUS
Telefax (61-2) 327 61 76

BRAZIL

05413 SÃO PAULO
R Henrique Schaumann 286-CJ33
Tel (55-11) 883-5455
Telex (39-11) 37988 "UMBR BR"

CANADA

BRAMPTON, ONTARIO
341 Main St North
Tel (416) 455-0505
Telefax 416-455-2606

CHINA

BEIJING
Beijing No 5 Semiconductor
Device Factory
14 Wu Lu Tong Road
Da Shang Mau Wai
Tel (861) 2024378
Telex 222722 STM CH

DENMARK

2730 HERLEV
Herlev Torv, 4
Tel (45-2) 94 85 33
Telex 35411
Telefax (45-2) 948694

FRANCE

94253 GENTILLY Cedex
7 - avenue Gallieni - BP 93
Tel (33-1) 47 40 75 75
Telex 632570 STMHQ
Telefax (33-1) 47 40 79 10

67000 STRASBURG

20, Place des Halles
Tel (33) 88 25 49 90
Telex 870001F
Telefax (33) 88 22 29 32

HONG KONG

WANCHAI
22nd Floor - Hopewell centre
183 Queen's Road East
Tel (852-5) 8615788
Telex 60955 ESGIES HX
Telefax (852-5) 8656589

INDIA

NEW DELHI 110 001
Lason Office
c/o Diners Business Services Pvt Ltd
World Trade Tower - First Floor
Barakhamba Lane
Tel 331 4668 - 331 2840
Telex 031 63421 DBSD IN
Telefax 331 2830

ITALY

20090 ASSAGO (MI)
V.le Milanofiori - Strada 4 - Palazzo A/4/A
Tel (39-2) 89213 1 (10 linee)
Telex 330131 - 330141 SGSAGR
Telefax (39-2) 8250449

40033 CASALECCHIO DI RENO (BO)

Via R. Fucini, 12
Tel (39-51) 591914
Telex 512442
Telefax (39-51) 591305

00161 ROMA

Via A. Torlonia, 15
Tel (39-6) 8443341/2/3/4/5
Telex 620653 SGSATE I
Telefax (39-6) 8444474

JAPAN

TOKYO 108
Nisseki - Takanawa Bld 4F
2-18-10 Takanawa
Minato-Ku
Tel (81-3) 280-4121
Telefax (81-3) 280-4131

KOREA

SEOUL 121
8th floor Shinwon Building
823-14, Yuksam-Dong
Kang-Nam-Gu
Tel (82-2) 552-0399
Telex SGSKOR K29998
Telefax (82-2) 552-1051

NETHERLANDS

5612 AM EINDHOVEN
Dillenburgstraat 25
Tel (31-40) 550015
Telex 51186
Telefax (31-40) 528835

SINGAPORE

SINGAPORE 2056
28 Ang Mo Kio - Industrial Park 2
Tel (65) 4821411
Telex RS 55201 ESGIES
Telefax (65) 4820240

SPAIN

08021 BARCELONA
Calle Platon, 6 4th Floor, 5th Door
Tel (34-3) 2022017-2020316
Telefax (34-3) 2021461

28027 MADRID

Calle Albacete, 5
Tel (34-1) 4051615
Telex 27060 TCCEE
Telefax (34-1) 4031134

SWEDEN

S-16421 KISTA
Borgarfjordsgatan, 13 - Box 1094
Tel (46-8) 7939220
Telex 12078 THSWS
Telefax (46-8) 7504950

SWITZERLAND

1218 GRAND-SACONNEX (GENÈVA)

Chemin Francois-Lehmann, 18/A
Tel (41-22) 7986462
Telex 415493 STM CH
Telefax (41-22) 7984869

TAIWAN

TAIPEI
12th Floor
571, Tun Hua South Road
Tel (886-2) 755-4111
Telex 10310 ESGIE TW
Telefax (886-2) 755-4008

UNITED KINGDOM and EIRE

MARLOW, BUCKS
Planar House, Parkway
Globe Park
Tel (44-628) 890800
Telex 847458
Telefax (44-628) 890391

U.S.A.

NORTH & SOUTH AMERICAN
MARKETING HEADQUARTERS
1000 East Bell Road
Phoenix, AZ 85022-2699
(1)-(602) 867-6340

SALES COVERAGE BY STATE

ALABAMA

Huntsville - (205) 533-5995

ARIZONA

Phoenix - (602) 867-6340

CALIFORNIA

Irvine - (714) 250-0455
San Jos - (408) 452-8585

COLORADO

Boulder (303) 449-9000

GEORGIA

Norcross - (404) 242-7444

ILLINOIS

Schaumburg - (708) 517-1890

MARYLAND

Columbia - (301) 995-6952

MASSACHUSETTS

Waltham - (617) 890-6688

NEW JERSEY

Voorhees - (609) 772-6222

OREGON

Tigard - (503) 620-5517

TEXAS

Austin - (512) 339-4191
Carrollton - (214) 466-8844

WASHINGTON

Seattle - (206) 524-6421

FOR RF AND MICROWAVE
POWER TRANSISTORS CONTACT
THE FOLLOWING REGIONAL
OFFICES IN THE U S A

CALIFORNIA

Hawthorne - (213) 675-0742

NEW JERSEY

Totowa - (201) 890-0884

PENNSYLVANIA

Montgomeryville - (215) 362-8500

TEXAS

Carrollton - (214) 466-8844

WEST GERMANY

6000 FRANKFURT

Gutleutstrabe 322
Tel (49-69) 237492
Telex 176997 689
Telefax (49-69) 231957
Teletex 6997689=STVDISTR

8011 GRASBRUNN

Bretonischer Ring 4
Neukeferloh Technopark
Tel (49-89) 460060
Telex 528211
Telefax (49-89) 4605454
Teletex 897107=STDISTR

3000 HANNOVER 1

Eckenerstrasse 5
Tel (49-511) 634191
Telex 175118418
Teletex 5118418 csfbeh
Telefax (49-511) 633552

8500 NÜRNBERG 20

Erlenstegenstrasse, 72
Tel (49-911) 597032
Telex 626243
Telefax (49-911) 5980701

5200 SIEGBURG

Frankfurter Str 22a
Tel (49-2241) 660 84-86
Telex 889510
Telefax (49-2241) 67584

7000 STUTTGART

Oberer Kirchhaldenweg 135
Tel (49-711) 692041
Telex 721718
Telefax (49-711) 691408

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of SGS-THOMSON Microelectronics.

© 1988 SGS-THOMSON Microelectronics — Printed in Italy — All Rights Reserved

* GAL and ispGAL are registered trademarks of Lattice Semiconductor Corporation,
E²CMOS, Generic Array Logic and PALtoGAL are trademarks of Lattice Semiconductor Corporation,
IFL is trademark of Signetics, PAL, PALASM are registered trademarks of Monolithic Memories, Inc.,
CUPL is a trademark of Personal CAD Systems Inc., ABEL, PLD test and DASH-ABEL is trademark of Data I/O Corporation
FastMap Plus, LC9000 are trademarks of PLT Inc
PLAQ is a trademark of QWERTY Inc

SGS-THOMSON Microelectronics GROUP OF COMPANIES

Australia - Brazil - China - France - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - The Netherlands -
Singapore - Spain - Sweden - Switzerland - Taiwan - United Kingdom - U.S.A. - West Germany



LOGMAST1

