MOTOROLA
Semiconductor Products Inc.

AN-824A
Application Note

MC68000 DMA USING THE MC6844 DMA CONTROLLER

The MC6844 DMA Controller (DMAC) can be interfaced to the MC68000 microprocessor to provide flexible, low-cost, relatively high performance DMA control in an MC68000-based system. In designing such a system, three interface requirements must be considered:

1. The DMAC should operate at maximum frequency for efficient data transfer. High performance systems may require the use of the two megahertz device (MC68B44), so the system must allow the MC68000 to access the DMAC asynchronously.
2. Handshake logic must be implemented to arbitrate control of the system bus between the MC68000, the DMA control system, and other possible bus masters.
3. The MC6844 is an 8-bit device intended for use in M6800 systems, capable of direct memory access through only a 64K memory space, and also lacks certain bus strobes necessary for simple implementation in an MC68000-based system. A bus interface must be designed to allow direct memory access throughout the entire 16 megabyte MC68000 memory map and to provide the required bus strobes needed for successful use in an MC68000-based system.

This application note describes designs to meet each of these requirements. These designs are then combined to form a direct memory access control system for the MC68000. An implementation of the complete system is presented in block diagram form using an MC6854 Advanced Data Link Controller (ADLC) and a static memory buffer.

MC6844 ASYNCHRONOUS INTERFACE OPERATION

The MC6844 can be interfaced asynchronously to the MC68000 using the circuitry presented in Figure 1. This circuit allows the MC68000 to access a DMAC driven by an E clock that is either synchronous or asynchronous to the MC68000 clock. It generates DMAC chip select at the proper time to satisfy DMAC timing requirements, latches data to satisfy data hold time requirements, and asserts data transfer acknowledge at the proper time to ensure valid data transfer between devices. This circuit can be used to interface other MC6800 peripherals, and is used to interface to the ADLC as well as the DMAC in the system implementation presented at the end of this application note.

CIRCUIT OPERATION — When the MC68000 performs a read or write bus cycle (access), the processor asserts one or both of the two data strobes ($\overline{DS}$), an address strobe ($\overline{AS}$), the read/write (R/$\overline{W}$) signal, and an address. The processor also outputs data during write cycles.

The MC68000 remains in this state until the bus cycle is terminated. Data transfer acknowledge ($\overline{DTACK}$) is asserted by the peripheral or memory device being accessed to initiate termination of the bus cycle by the MC68000.

The circuit in Figure 1 synchronizes MC68000 accesses to the DMAC with the E clock. Initially, flip-flops U1A and U1B are cleared causing a high $\overline{DTACK}$ output setting U2 and U3 into a transparent mode. Latch U2 is in the high-impedance state due to a high on the output enable ($\overline{OE}$) input. Latch U3 is enabled due to a low on the $\overline{OE}$ input.

At the start of a DMAC access, latch U3 remains enabled if the access is a write. If the access is a read, the high R/$\overline{W}$ and DMAC Select inputs to U4A cause U3 to go to the high-impedance state and U2 to become enabled. The DMAC Select signal is asserted when the DMAC is addressed. However, the DMAC is actually selected by the assertion of CS (DMAC). Flip-flop U1A is clocked high on the first falling edge of E after DMAC Select and data strobe (DS) are asserted. The Q output of U1A is applied to U4D, asserting CS (DMAC). Selecting the DMAC at this time ensures that the DMAC has adequate address setup time.

On the next falling edge of E, the $\overline{Q}$ output of U1B is clocked low asserting $\overline{DTACK}$ and latching data into the enabled latch. The asserted $\overline{DTACK}$ signal, inverted by U4D, deselects the DMAC by causing $\overline{CS}$ (DMAC) to go high. When the access terminates, flip-flop U1 is cleared by the negation of $\overline{DS}$, and the interface circuitry is initialized for the next access. The $\overline{DTACK}$ signal is buffered by an open-collector buffer (U5) to allow assertion of $\overline{DTACK}$ by other devices when the DMAC is not being accessed.

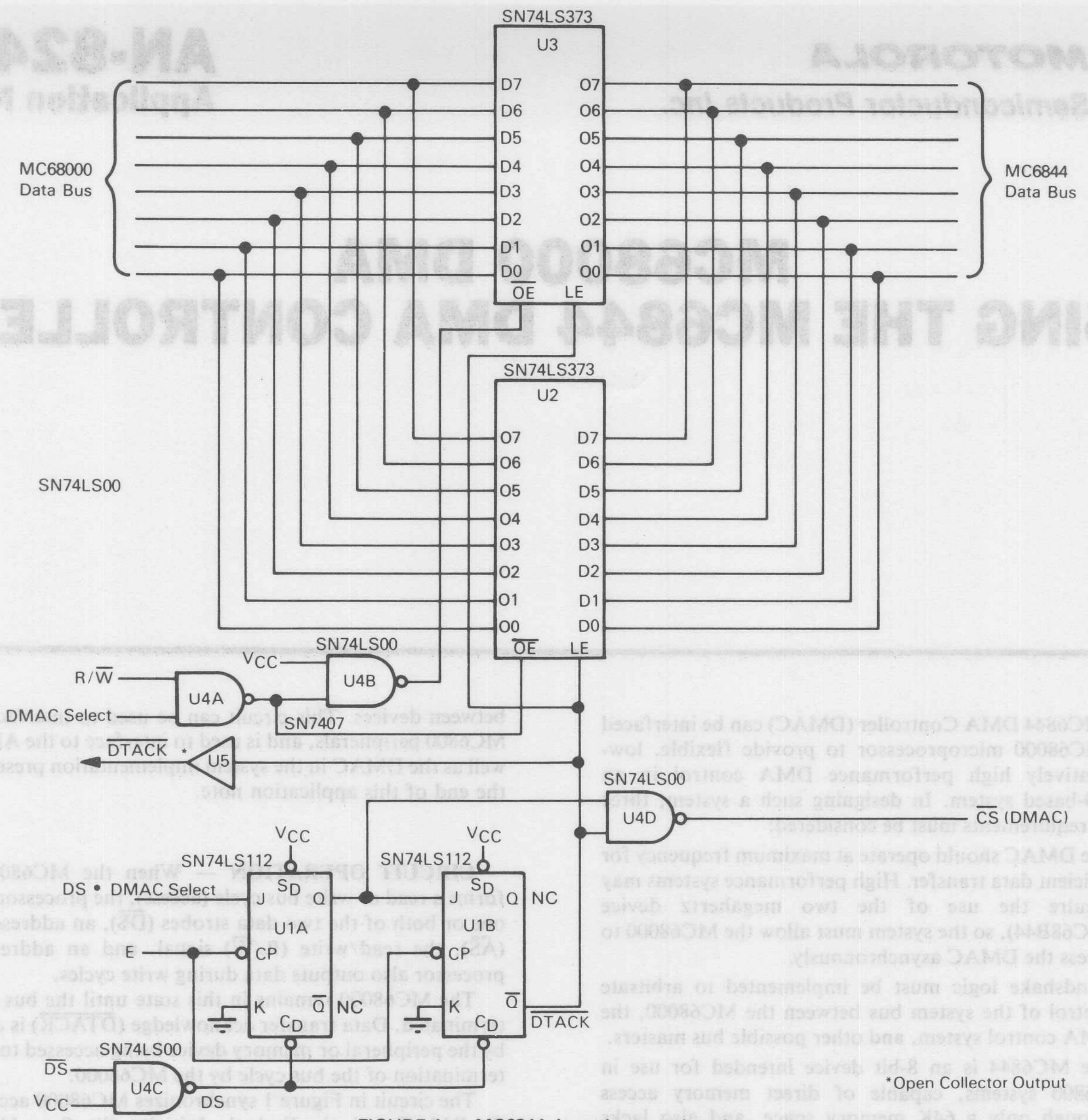

FIGURE 1 — MC6844 Asynchronous Interface

BUS ARBITRATION INTERFACE

The MC6844 is an 8-bit, 4-channel DMA Controller capable of performing direct memory transfers of a user defined number of data bytes (data block) within a 64K byte memory space. Associated with each channel of the controller are:

- A transfer request (TxRQ) input which is asserted by a peripheral controller or a processor to request DMA service.
- A 16-bit address register which is initialized with the beginning address of the data block to be transferred.
- A 16-bit byte count register which is initialized with the desired number of data bytes (size of the data block) to be transferred.

Each channel can perform DMA transfers in one of three modes: TSC Steal, Halt Steal, and Halt Burst. Two of these modes, TSC Steal and Halt Steal, are single-byte transfer modes in which the DMAC returns control of the system bus to the processor after each transfer, while the Halt Burst mode is a block transfer mode in which the DMAC retains control of the system bus until the last byte of the data block has been transferred.

The bus arbitration circuit presented in Figure 2 is designed for the Halt Steal and Halt Burst modes of operation. The TSC Steal mode is intended for use with the MC6800 and offers no advantage over the Halt Steal mode in MC68000 applications.

In the Halt Steal mode the DMAC responds to a transfer request by asserting DMA request halt steal ($\overline{\text{DRQH}}$). The DMAC then waits until DMA grant (DGRNT), a DMAC input, is asserted. At this time, one transfer of data is initiated and transfer strobe ($\overline{\text{TxSTB}}$) is asserted, followed by the negation of $\overline{\text{DRQH}}$. This sequence is repeated until all data has been transferred.

The same sequence is followed in the Halt Burst mode with the exception that $\overline{\text{DRQH}}$ is negated only after the last byte of the data block has been transferred. In this mode, bus mastership is arbitrated once, then data transfers occur in succession until all data has been transferred.

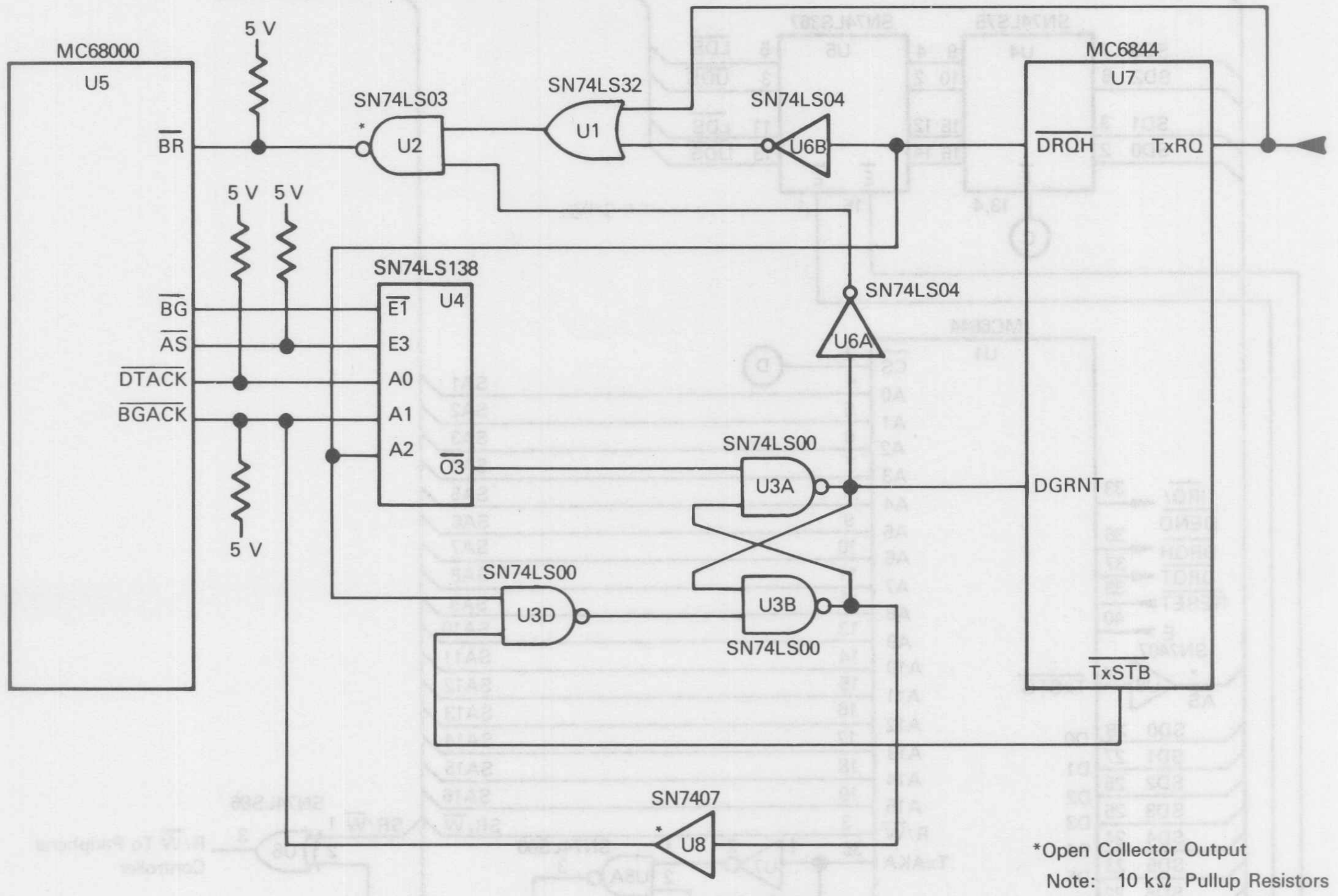

FIGURE 2 — Bus Arbitration Interface

CIRCUIT OPERATION — For either a Halt Steal or Halt Burst DMA transfer by the control systems presented in this application note, three conditions must be met:

1. Transfer request (TxRQ) must be asserted.
2. $\overline{\text{DRQH}}$ must be asserted.
3. All bus masters must have relinquished the bus to ensure that DMA grant (DGRNT) is asserted.

Initially DGRNT is low, bus grant acknowledge ($\overline{\text{BGACK}}$) is not asserted by the interface, and $\overline{\text{TxSTB}}$ is high. The DMAC responds to a transfer request by asserting $\overline{\text{DRQH}}$. Once $\overline{\text{DRQH}}$ is asserted, it remains asserted until the DMAC performs a byte transfer in the Halt Steal mode or until the last byte of a designated memory block is transferred in the Halt Burst mode.

Transfer request (TxRQ) is coupled through U1 and U2 so that MC68000 bus request ($\overline{\text{BR}}$) is asserted when TxRQ is asserted. By requesting a DMA transfer and bus arbitration simultaneously (disregarding gate propagation delay), DMA latency time is minimized. The MC68000 responds to a bus request by asserting bus grant ($\overline{\text{BG}}$) and relinquishing the bus.

When DRQH is asserted and all bus masters are off the system bus, indicated by the negation of $\overline{\text{AS}}$, $\overline{\text{DTACK}}$, and $\overline{\text{BGACK}}$, flip-flop U3A-U3B is set by the assertion of the $\overline{\text{O3}}$ output of U4. The setting of flip-flop U3A-U3B asserts DGRNT to initiate DMA transfer(s), and also asserts $\overline{\text{BGACK}}$ to keep other bus masters off the bus. Bus grant ($\overline{\text{BG}}$) is negated by the MC68000 soon after $\overline{\text{BGACK}}$ is asserted.

Flip-flop U3A-U3B is cleared on the rising edge of $\overline{\text{TxSTB}}$ after it is asserted during each DMA cycle in the Halt Steal mode, and during the last cycle of a block transfer in the Halt Burst mode. Clearing flip-flop U3A-U3B negates $\overline{\text{BGACK}}$ to release the system bus, and negates DGRNT to stop DMAC transfer activity.

The MC68000 $\overline{\text{BR}}$ and $\overline{\text{BGACK}}$ signals are driven by open collector gates to allow other devices to also request the system bus. A pullup resistor is used to hold $\overline{\text{AS}}$ in the negated state during transitions in bus ownership.

BUS INTERFACE REQUIREMENTS

A general direct memory access controller for an MC68000-based system must allow direct memory access throughout the entire 16 megabyte memory map of the MC68000. In addition, it must assert the appropriate data strobe(s) and an address strobe. The MC6844 does not satisfy these requirements; therefore, TTL devices must be used to meet these needs.

The MC68000 can perform three types of data transfers: word transfers (D0-D15), byte transfers to/from lower data bytes (D0-D7), and byte transfers to/from upper data bytes (D8-D15). When transferring a byte, the MC68000 asserts either the upper data strobe ($\overline{\text{UDS}}$) or the lower data strobe ($\overline{\text{LDS}}$), depending on whether an upper or a lower data byte is being transferred; and when transferring a word, it asserts

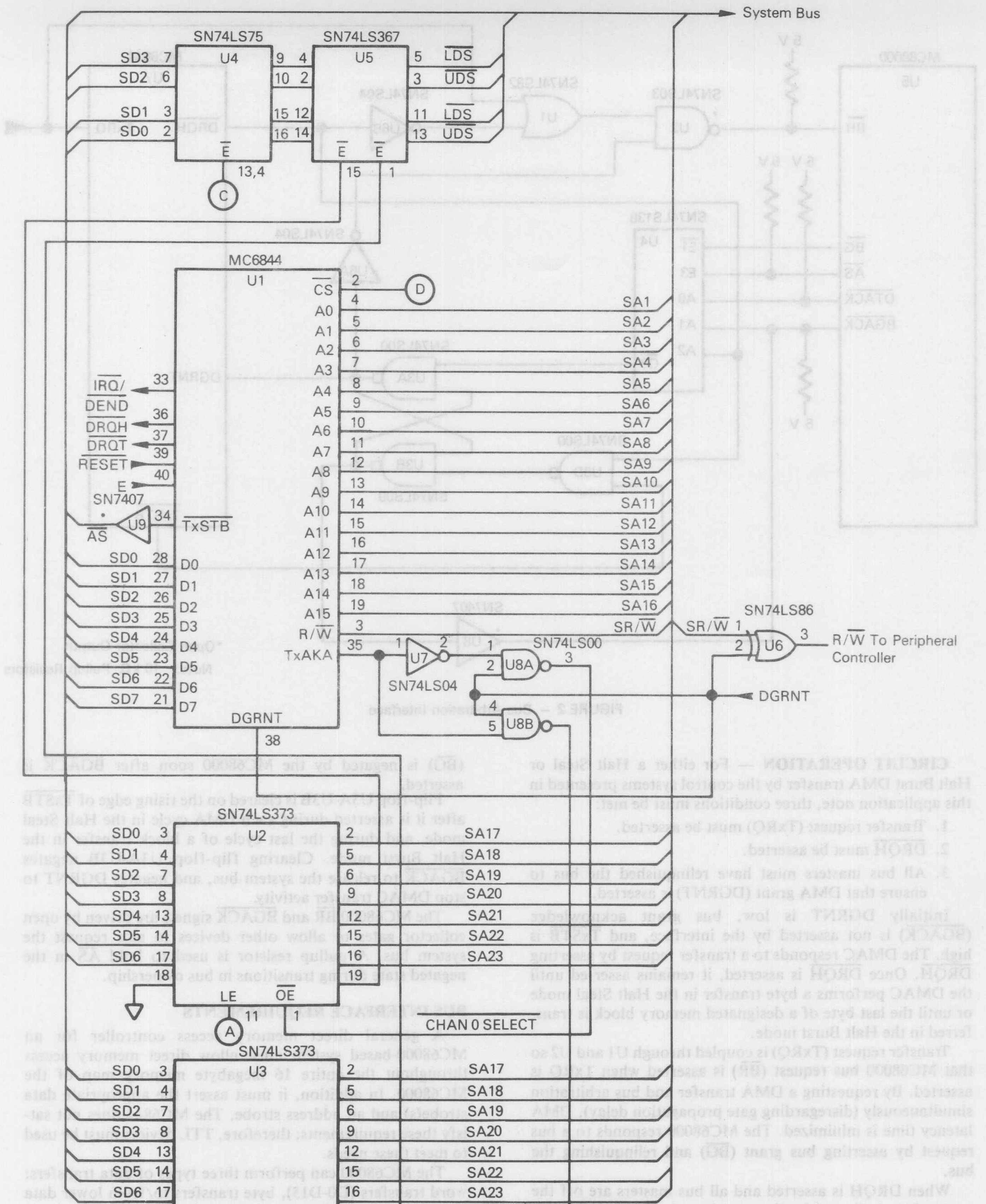

FIGURE 3 — MC68000 Word and Non-Sequential Byte Transfer Interface System

both $\overline{\text{UDS}}$ and $\overline{\text{LDS}}$. The MC68000 asserts $\overline{\text{AS}}$ during each type of transfer.

The following are general designs which can be modified to meet individual system requirements. The two designs presented differ in the types of transfer they perform.

Only two of the four DMAC channels are used in each design. However, these interfaces can be easily modified for four-channel operation.

WORD AND NON-SEQUENTIAL BYTE TRANSFER INTERFACE SYSTEM

An MC68000 DMA control system capable of word transfers and byte transfers to/from upper byte or lower byte memory locations is presented in Figure 3.

In this system, address lines A0-A15 from the DMAC are connected to MC68000 system address lines SA1-SA23 and as the DMAC address lines increment or decrement (according to user option), the system address is incremented/decremented by words, rather than bytes; that is, the system address changes in increments of two bytes.

The system upper address lines SA17-SA23, are latched into transparent latches U2 and U3 during initialization, which are enabled during a DMA transfer. Latch U2 is the channel 0 upper address latch, with its chip select labeled A; latch U3 is the channel 1 upper address latch, with its chip select labeled B. During a direct memory access, transfer acknowledge A (TxAKA) from the DMAC is asserted during channel 1 transfers, and negated during channel 0 transfers. This DMAC output is used to enable the proper address latch during a direct memory access.

The type of direct memory access transfer (word or byte) is determined by the state of latch U4 during the access. Latch U4 with its chip select labeled C, is connected to system data bus lines SD0-SD3 and, through three-state buffer U5, to system data strobes $\overline{\text{LDS}}$ and $\overline{\text{UDS}}$. When writing to latch U4 during initialization, the states of SD2 and SD3 determine the states of the data strobes during a channel 1 direct memory access, and the states of SD0 and SD1 determine the states of the data strobes during a channel 0 direct memory access. For word transfer both of the data strobes must be asserted, while for byte transfers either the $\overline{\text{LDS}}$ or $\overline{\text{UDS}}$ is asserted, depending on whether a lower data byte (D0-D7) or an upper data byte (D8-D15) is being transferred.

Note that in memory organized in 16-bit words, byte transfers are to/from either the upper byte or the lower byte of memory during each DMA block transfer.

During a direct memory access the appropriate U4 latch states are gated onto the system bus by U5. The appropriate U5 buffers are enabled by latch U2 during channel 0 access, and by latch U3 during channel 1 access.

When $\overline{\text{DGRNT}}$ is asserted, the R/$\overline{\text{W}}$ signal to the peripheral controller is inverted by exclusive OR gate U6.

Transfer strobe ($\overline{\text{TxSTB}}$) is fed through an open collector buffer to the system $\overline{\text{AS}}$ line. During a direct memory access transfer the $\overline{\text{AS}}$ output of the MC68000 is in the high-impedance state and $\overline{\text{TxSTB}}$ is used as the system address strobe. Transfer strobe is asserted by a DMAC operating at 2 megahertz for at least 370 nanoseconds to indicate a valid address during a direct memory access, and may require conditioning for use as an address strobe during direct memory access in some systems.

SEQUENTIAL MEMORY BYTE TRANSFER INTERFACE SYSTEM

An MC68000 DMA control system capable of byte transfers to/from sequential memory locations in a memory organized in 16-bit words is presented in Figure 4.

In this system, address lines A5-A15 from the DMAC are connected to MC68000 system address lines SA5-SA15. During a direct memory access, address lines A1-A4 from the DMAC are connected to MC68000 system address lines SA1-SA4 through buffer U9. Address line A0 from the DMAC is connected through inverters U4 to generate the data strobes.

Only one data strobe is asserted at a time. Each time the DMAC increments/decrements, the state of $\overline{\text{UDS}}$ and $\overline{\text{LDS}}$ alternate. System address line SA1 changes state only after each data strobe is asserted for one DMA cycle and negated for one DMA cycle. By doing this, data is transferred to/from consecutive byte locations in the word-dimensioned memory map.

When the MPU has to access the DMAC, buffer U8 is enabled by $\overline{\text{CS}}$ and address lines A0-A4 are connected to MC68000 system address lines SA1-SA5.

Latches U2 and U3 latch upper system address lines SA16-SA23 during initialization and their operation is identical to the circuit presented in Figure 3.

COMPLETE SYSTEM IMPLEMENTATION

A block diagram of a complete MC68000 DMA system using the MC6844 DMAC for controlling DMA between an MC6854 ADLC and a block of memory is presented in Figure 5. Data transfer in this system is between the ADLC and lower memory byte locations (D0-D7).

The ADLC asserts receiver data service request (RDSR) each time the receiver FIFO register requires servicing, and transmitter data service request (TDSR) each time the transmitter is ready for data. These outputs are tied to transfer request channel 0 (TxRQ0) and transfer request channel 1 (TxRQ1) of the DMAC so that DMAC channel 0 services the ADLC receiver, and DMAC channel 1 services the ADLC transmitter.

The block labeled "MC6854 Register Select, R/$\overline{\text{W}}$ Control" is used to address the ADLC transmit or receive register and to invert the read/write signal during a direct memory access. This circuit puts the address bus from the ADLC in the high-impedance state during the direct memory access and forces ADLC register select zero (RS0) to a low state and register select one (RS1) to a high state, so that during a direct memory access either the transmit FIFO register or the receiver FIFO register is selected according to the state of the R/$\overline{\text{W}}$ signal. The circuit uses DMAC $\overline{\text{DEND}}$ to select the frame terminate register of the ADLC during the last byte of a DMA block transfer when servicing the transmitter FIFO. During a direct memory access, the ADLC is selected by assertion of $\overline{\text{TxSTB}}$ to ensure that the ADLC is selected only during valid direct memory access cycles.

System memory is connected directly to the system bus. The "Memory $\overline{\text{DTACK}}$ Gen." consists of a counter driven by the MC68000 clock, and enabled by an asserted address strobe when memory is accessed by the processor. Data transfer acknowledge ($\overline{\text{DTACK}}$) is "picked off" one of the counter pins so that it is asserted at some preset time interval after memory is accessed.

Memory address decoding is the same for both direct

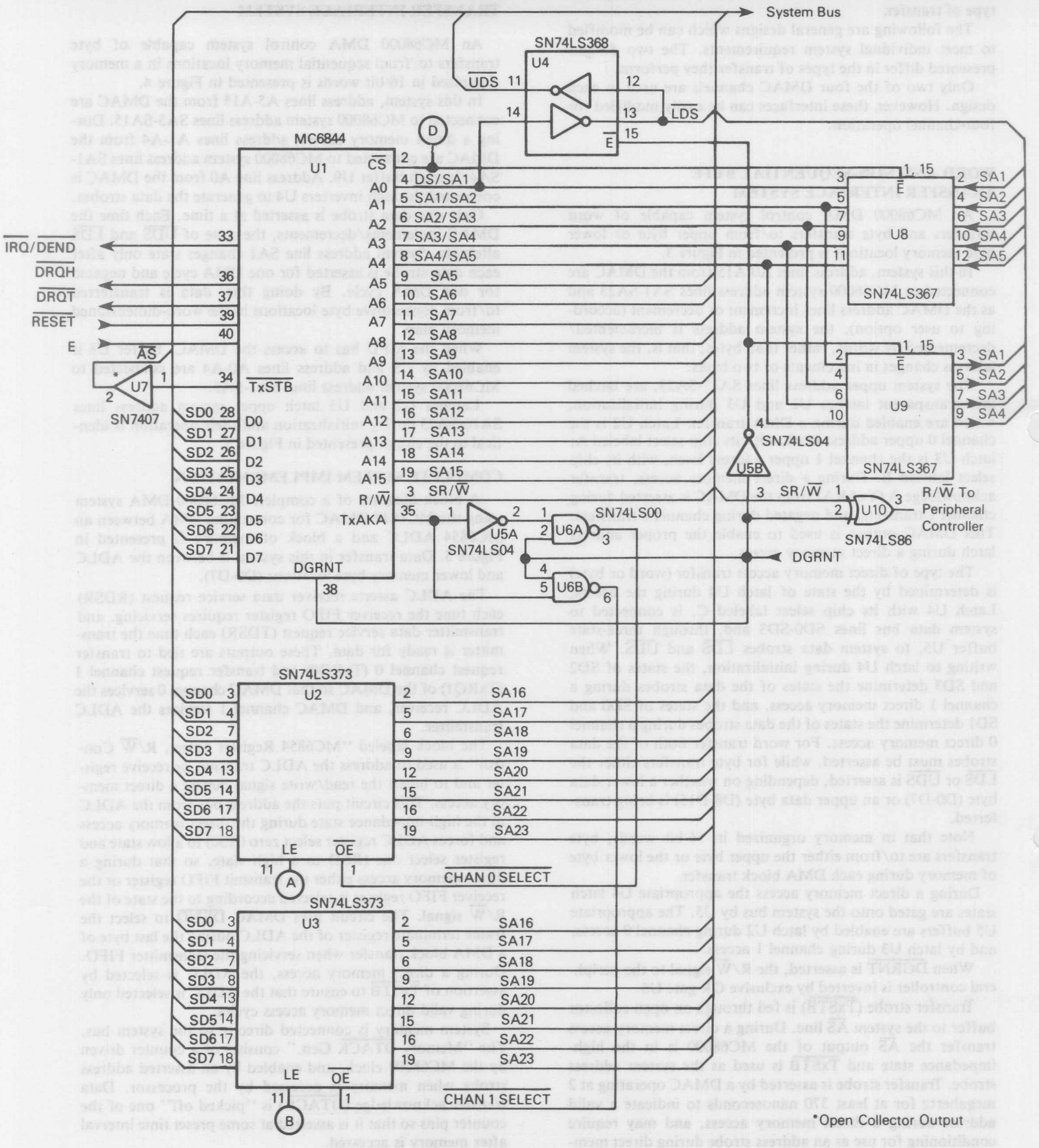

FIGURE 4 — MC68000 Sequential Memory Byte Transfer Interface System

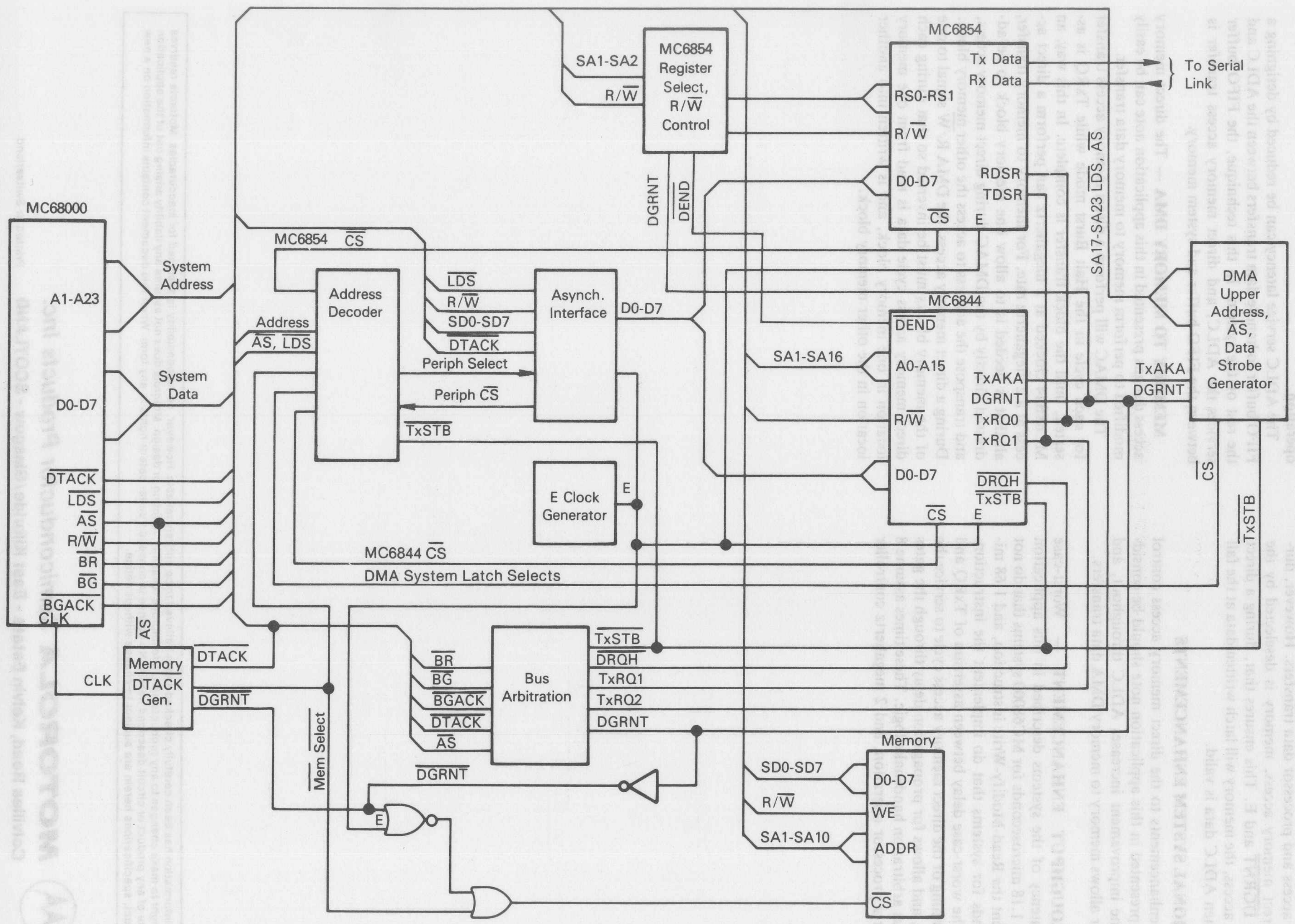

FIGURE 5 — System Implementation

memory access and processor data transfers. However, during a direct memory access, memory is deselected by the NOR or $\overline{\text{DGRNT}}$ and E. This ensures that, during a direct memory access, the memory will latch written data at the fall of E, when ADLC data is valid.

ADDITIONAL SYSTEM ENHANCEMENTS

Two enhancements to the direct memory access control systems presented in this application note should be considered. One improvement increases ADLC throughout, and the other allows memory to memory DMA data transfers.

THROUGHPUT ENHANCEMENT — Worst-case DMA latency of the systems described in this application note are 1.18 microseconds for MC68000 systems that do not implement the Read-Modify-Write instruction, and 1.68 microseconds for systems that do implement the instruction. This is the worst-case delay between assertion of TxRQ and the beginning of the direct memory access cycle to service the channel, and allows for propagation delay through the gates in the bus arbitration handshake logic. These times assume 8 megahertz processor operation, and 2 megahertz controller operation.

The ADLC service latency can be reduced by designing a FIFO buffer to handle data transfers between the ADLC and the rest of the system. In this technique, the FIFO buffer services the ADLC, and direct memory access transfer is between the FIFO buffer and system memory.

MEMORY TO MEMORY DMA — The direct memory access designs presented in this application note can be easily modified to perform memory to memory data transfer.

The DMAC will perform a direct memory access transfer for each cycle in the Halt Burst mode while TxRQ is asserted, until the block transfer is complete. In this way, an MC68B44 clocked at 2 megahertz can perform a direct access at a 2 megahertz rate. For memory to memory transfer, all that is needed is to allow one memory block to be addressed directly by the DMAC during direct memory access, and transpose the address to access the other memory block. During a direct memory access, the DMA R/$\overline{\text{W}}$ signal to one of the memory blocks must be inverted so that during each direct memory access cycle data is read from one memory location in one memory block, and is written into another location in the other memory block.

MOTOROLA Semiconductor Products Inc.
Colvilles Road, Kelvin Estate - East Kilbride/Glasgow - SCOTLAND
Printed in Switzerland