COMPUTER SYSTEMS LABORATORY
WASHINGTON UNIVERSITY
ST. LOUIS, MO. 63110

LINC Document No. 35 August 5, 1968

Programming the LINC
Appendix III: LINC Modifications

M. J. Stucki and M. L. Pepper

Abstract

In August 1965 an interrupt feature, the Z Register, and five new instructions were made available on the LINC.

The Computer System Laboratory is supported in part by the Advanced Research Projects Agency of the Department of Defense under contract SD-302, and in part by the Division of Research Facilities and Resources of the National Institutes of Health under grant FR-00396.

LINC ARITHMETIC EXTENSIONS

The Z Register: This 12-bit register can be thought of as being to the right of the Accumulator. It is used as a utility register with the DSC and SAM instructions, and it holds the least significant half of the product following a MUL instruction. Each shift of the Accumulator during ROR and SCR also shifts the contents of the Z Register right with $A_0 \rightarrow Z_{11}$. (ROR 14 transfers C(ACC) to Z.) The Z Register is cleared by CLR. MUL, DSC, SAM, ROR, SCR, and CLR are the only instructions which alter the contents of the Z Register.

Following MUL, the least significant 11 bits of the product are in Z_1 through Z_{11}. Though the half product in the Accumulator is left with the correct sign, the half in the Z register is always positive. Since the sign is left in the Link bit, the following will recover the least significant half as an 11-bit signed number:

```
ZTA
LZE
COM
```

The most significant 11 bits are lost if an integer multiplication is executed.

Overflow. The following instructions set the overflow flag: ADD, ADA, ADM, and LAM. If there is overflow during execution of one of these instructions the overflow flag is set on, if there is no overflow, it will be set off. Overflow results when two numbers of the same sign are added and the sum is of the opposite sign.

M. L. Pepper
C. S. L. 8/5/68

LINC INTERRUPT

A feature has been added to the LINC which permits a program to be interrupted in the course of its operation. This feature has no effect until activated by a special interrupt enable instruction, ENI (MSC 10). Thereafter, if an Interrupt request occurs, the normal running of the program will be interrupted and the next instruction will automatically be taken from location 21. Two kinds of interrupt, a program interrupt and a data interrupt are available. Which one of these will occur depends on the instruction in location 21.

DATA INTERRUPT: Data interrupts are used to transfer data between memory and an external piece of equipment. This is done by putting an OPR instruction in register 21 and executing it in the GULP mode. The BCOM operation normally performed at 2.2 time of an OPR is inhibited so that the Accumulator will not be affected unless it is intentionally disturbed by the assertion of CLEL, SNEL, or TNEL. At the end of the OPR instruction, the machine will resume running the interrupted program.

PROGRAM INTERRUPT: A program interrupt allows the program to execute a special routine (service routine) whenever an interrupt occurs. This routine may be located anywhere in memory; it may not, however, begin in locations zero or 21. To arrange for a program interrupt, one puts the instruction "JMP X" in register 21 (X being the address of the service routine). This accomplishes three things:

1. It transfers program control to the service routine.
2. It stores the instruction "JMP n" in register zero (n is the address of the next instruction in the interrupted program).
3. It disables the interrupt feature so that the machine cannot be interrupted during the service routine.

REQUESTING AN INTERRUPT: A -3V level on the pin called INTREQ (FC30) will request an interrupt. The level may occur asynchronously with the main machine but it must remain until the interrupt actually occurs. At that time a -3V level will appear on the pin called BDOINTFF[1] (FC15), indicating that the instruction in register 21 is being executed. The interrupt request must be removed within 16 usec of the time this level appears.

WHERE INTERRUPTS CAN OCCUR: If the interrupt mode has been activated and an interrupt request appears, the program will be interrupted as soon as one of the following occurs:

1. The end of a non-JMP instruction. A program cannot be interrupted at the end of a JMP instruction.

2. The end of a non-ENI instruction. A program cannot be interrupted at the end of the instruction ENI.

   NOTE: This assumes that the interrupt feature is being activated by the ENI. However, if the interrupt feature is already active, i. e., the ENI is redundant, an interrupt can occur at the end of the instruction.

3. The occurence of a pause. An MTP or OPR instruction can be interrupted during the paused state. The instruction will be terminated abruptly and the interrupt executed. At the end of the interrupt the machine will return to the next instruction; it will not return to the unfinished instruction.

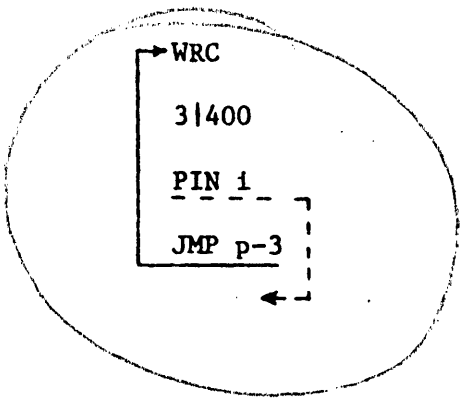
WRITING INTERRUPTABLE PROGRAMS: Programs utilizing the interrupt feature must be specially written in any section that can be interrupted.

1. Programs incorporating a program interrupt: The very first instruction in each subroutine must save the contents of register zero. This is necessary since a program interrupt occurring during the subroutine will destroy the contents of register zero.

   NOTE: An interrupt cannot occur immediately before the first instruction in a subroutine since that instruction is preceded by a JMP.

2. Programs incorporating either interrupt: Whenever an instruction is interrupted in the paused state, a flip-flop called PINFF (Pause Interrupt Flip-Flop) is set to a one. The state of this flip-flop can be checked with the instruction PIN (SKP 6). The PINFF should be checked after every instruction that pauses and the instruction should be repeated if an interrupt occurred.

Example:



WRITING SERVICE ROUTINES:

1. If a service routine uses

   A. the Accumulator: the initial contents of the Accumulator must be saved and restored to it at the end of the routine

   B. a JMP instruction: the return JMP in register zero must be saved. 2 REG

2. The interrupt feature is automatically disabled upon entering a service routine. If the interrupt feature is to be operative upon returning to the interrupted program, the service routine must reactivate it just prior to the return. The instruction ENI must be the very last instruction before the return JMP. If it occurs any earlier, the service routine itself may be interrupted.

DISABLING THE INTERRUPT MODE:

1. Manually: Pushing the STOP switch on the console disables the interrupt mode. It also clears the PINFF.

2. Programs incorporating a program interrupt: The interrupt mode is automatically disabled every time an interrupt occurs. If it is not re-activated by an ENI at the end of the service routine, it will remain disabled.

3. Programs incorporating either interrupt: Putting the instruction NOP in register 21 disables the interrupt mode.

   NOTE: This will not disable the interrupt mode until the next interrupt request occurs. At that time the NOP is executed and the interrupt mode disabled. At the end of the NOP, the machine resumes running the interrupted program.

4. The paused state can not be interrupted while the PINFF is set to a one.

M. J. Stucki
C. S. L. 8/18/65

Additional LINC Instructions (The following instructions were added to the LINC in 1965)

Miscellaneous Class

ZTA	0005	8 μsec.	ZTA
Z REGISTER TO ACCUMULATOR. Clear the Accumulator and then transfer the contents of the Z register to the Accumulator. The transfer is offset, so that $Z_i \rightarrow A_{i-1}$ and $0 \rightarrow A_{11}$. Z_0 is not transfered.			

ENI	0010	8 μsec.	ENI
ENABLE INTERRUPT. Enable the interrupt mode.			

SKIP CLASS

ZZZ i	455 + 20i	8 μsec.	ZZZ
Z ZERO ZERO. Condition: Bit zero of the Z Register contains 0.			

OVF i	454 + 20i	8 μsec.	OVF
OVERFLOW. Condition: The overflow flag is on. This instruction does not clear the overflow flag.			

PIN i	446 + 20i	8 μsec.	PIN
PAUSE INTERRUPTED. Condition: The PINFF (Pause Interrupted flip-flop) is set to a one. Execution of this instruction clears the flip-flop.			

M. L. Pepper
C.S.L. 8/5/68