

TECHNICAL NOTE 74

HOW TO USE INTERRUPTS

on the

2408 PROCESSOR

J. J. Schwertel
3-1-73

REV A

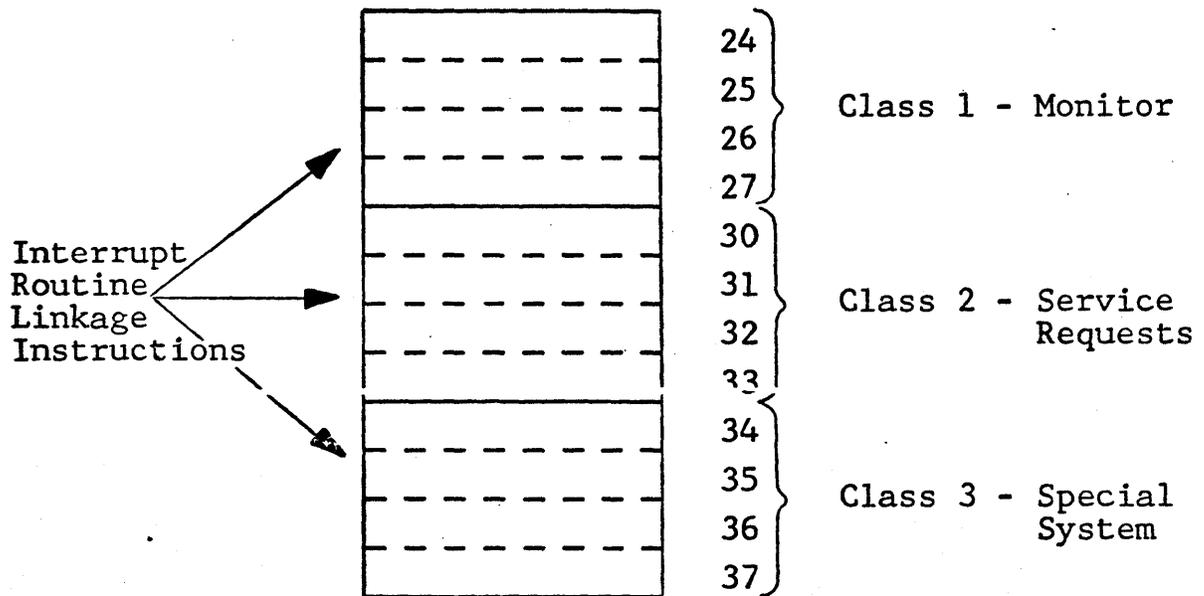
P R E F A C E

This technical note describes how to use 2408 Processor interrupts. It is intended to supply the users with enough information to write interrupt driven programs on the 2408 Processor.

1.0

GENERAL

The 2408 processor provides three classes of interrupts with a variable number of uniquely identifiable interrupts in each class. An interrupt forces the execution of an instruction at a fixed memory location associated with the particular interrupt class. The allocation of memory for interrupt linkage instructions is shown in the following diagram:



The interrupt linkage instruction would normally be a GOTO SUBROUTINE (GSB) instruction to the appropriate interrupt routine, thereby causing P to be saved in the push down stack specified in OPl. When an interrupt occurs, further interrupts are locked out until enabled by the program. The enabling of interrupts is accomplished by executing either a CLEAR INTERRUPT LOCKOUT (CIL) or an INTERRUPT RETURN GOTO (GIR) instruction. The CLEAR INTERRUPT LOCKOUT (CIL) instruction will enable interrupts that are not selectively locked out, to force the

execution of the instruction at the fixed memory location associated with the particular class of interrupts. Under a RESTART-RUN condition all interrupts are locked out and a CLEAR INTERRUPT LOCKOUT (CIL) instruction must be executed if interrupts are to be utilized by the program. If a condition occurs where a portion of a program must be run without interruption, the SET INTERRUPT LOCKOUT (SIL) is used to lock out all interrupts. Interrupts will then remain locked out until either a CLEAR INTERRUPT LOCKOUT (CIL) or an INTERRUPT RETURN GOTO (GIR) instruction is executed. Interrupts may be selectively enabled or disabled by the INTERRUPT MASK (IM) instruction.

Selective Lockout of Interrupts

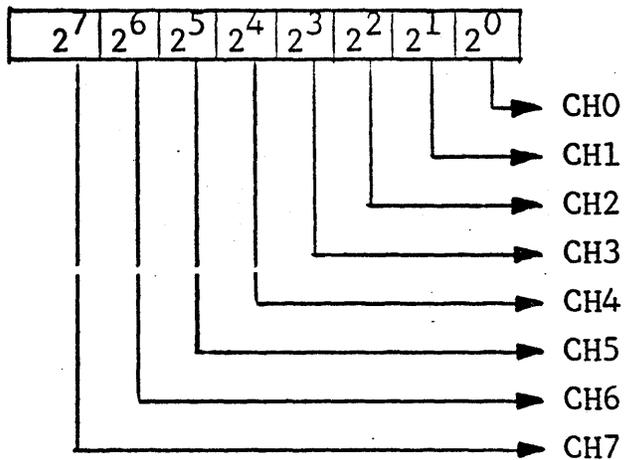
INTERRUPT MASK (IM) Instruction

OP Code	OPl
174	XXX

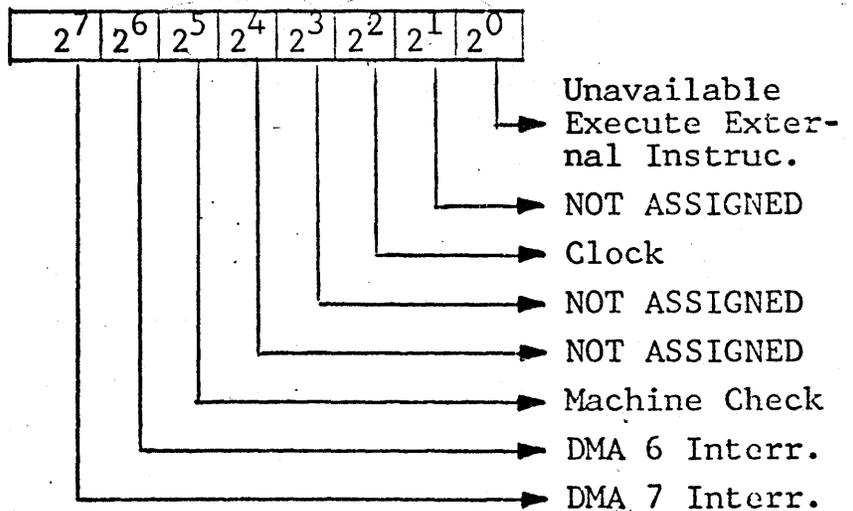
OPl is a three byte item with a 1 bit selectively locking out Interrupts

1st byte - Lockout Monitor Interrupts

2nd byte - Lockout Service Interrupts



3rd byte - Lockout Class III Interrupts



The occurrence of an interrupt forces an automatic state swap to the interrupt AR set, exec State. This set of AR's is located in core locations 040₈ through 057₈ and loaded into the corresponding hardware AR's when the interrupt occurs. This set of AR's will be used until the execution of a SWAP STATES (SWS) instruction. After the execution of a SWAP STATES (SWS) instruction, one more instruction will execute under the exec. state and then the worker state AR's will be activated. The normal exit from an interrupt subroutine would be the execution of a SWAP STATE (SWS) followed by an INTERRUPT RETURN GOTO (GIR) instruction. In addition to enabling interrupts, the GIR instruction will retrieve the return address from the push down stack specified in OPl, therefore the OPl items of the GSB and GIR instructions should be the same in order to return to the main program at the exit point.

Upon entering an interrupt subroutine, the STORE TALLY COUNTER (STT) and the STORE DESIGNATORS (STD) instructions are normally the first instructions to be executed prior to the instruction used to determine the cause of the interrupt. The purpose of these two instructions is to save the condition of the tally counter and designators as they are set prior to the interrupt. During the interrupt routine the tally and designators may be changed and used for sensing interrupt subroutine conditions. Prior to an interrupt routine exit, the LOAD DESIGNATORS (LD) and TALLY COUNTER (LT) instructions are executed to restore the designators and tally counter for the worker state.

A class 1 interrupt is generated when an I/O selector channel goes from an active state to an inactive state. This interrupt can be generated by any of the eight selector channels and a record of this event is saved by the associated channel interrupt flip-flop. When a monitor interrupt occurs it will force a jump to address 24 which will normally contain a GSB instruction to save P and generate a jump to the monitor interrupt subroutine. In the monitor interrupt subroutine, a series of GOTO ON CHAN. INTERRUPT (GCI) instructions can be used to determine which selector I/O channel caused the interrupt. This instruction also clears the interrupt condition. At the end of this subroutine an INTERRUPT RETURN GOTO (GIR) instruction is normally used to return to the main program. Class 1 interrupts may be selectively enabled or disabled in OPl, byte 1 of the INTERRUPT MASK (IM) instruction.

EXAMPLE MONITOR INTERRUPT

24)

GSB	B	MON	SUB
-----	---	-----	-----

	MON SUB)	STT AB
		STD AC
		GCT CH0 SUB 0
		GCT CH1 SUB 1
		GCT CH2 SUB 2
		GCT CH3 SUB 3
		GCT CH4 SUB 4
		GCT CH5 SUB 5
		GCT CH6 SUB 6
		GCT CH7 SUB 7
	MON SUB 1)	ID AR
		LT AC
		SWS
		GIR B

OPl Item of
GCT Instruction
 CH0 = 000
 CH1 = 001
 CH2 = 002
 CH3 = 003
 CH4 = 004
 CH5 = 005
 CH6 = 006
 CH7 = 007

When a monitor interrupt occurs, the GSB instruction at address 24 will store P in the push down stack designated by the B Item of OPl and causes a jump to address MON SUB. The STT instruction will store the tally counter in the first two bytes of the item specified by AB and the STD instruction will store the designators in the first byte of the item specified by AC. The GCT instructions will now compare the mask contained in the OPl item with the condition of the channel interrupt flip-flops. When the bit in the mask and the

channel interrupt flip-flop compare a jump will be made to the address + P Bias that is carried in the last two bytes of the GCT instruction. This jump takes the program to the subroutine designated to process the interrupt for that particular channel. At the end of this subroutine a jump is made to address MON SUB 1 which now will reload the tally counter and the designators before returning to the main program. The return to the main program is done by the GIR instruction which will retrieve the return address from the push down stack specified by B item in OPl.

3.0 CLASS 2 - SERVICE REQUESTS

A class 2 interrupt is generated by a device attached to an I/O selector channel. This interrupt can be generated on any of the eight selector channels and a record of this event is saved by the associated selector channel. The intended use for this interrupt class is to capture external conditions that require processor actions. In order to generate a service request, the peripheral device controller must interface with a service request line which is included in all I/O cables. If more than one interruption device is attached to a single channel, interface logic is responsible for capturing and retaining the service request until interrogated by an EF.

When a service interrupt occurs, it will force a jump to address 30 which will normally contain a GSB instruction to save P and generate a jump to the service interrupt subroutine. In the service interrupt subroutine a series of GOTO ON SERVICE REQUEST (GSI) instructions are used to identify the interrupting channel, similar to the subroutine used for monitor interrupts. If more than one device on a given channel is connected to the service request line, the interrupting device is identified by requesting status from each device. When status is requested from the interrupting device, the service request status bit will be set and the interrupt condition will be cleared. Class 2 interrupts are handled similar to monitor interrupts and can be selectively enabled or disabled in OP1, byte 2 of the INTERRUPT MASK (IM) instruction.

4.0 CLASS 3 - SPECIAL SYSTEM

Class 3 interrupts are generated by special internal conditions with eight class 3 interrupts sources being the maximum for the standard 2408. Class 3 interrupts are identified by executing an EXECUTE EXTERNAL INSTRUCTION (EXI) with a sub-operation code of STORE EXTERNAL INSTRUCTION ERROR (SEE) which has an octal code of 014. (See EXI Instruction Format.) Interrupt sources are bit position encoded in the second byte of OP3 as follows:

<u>BIT POSITION</u>	<u>INTERRUPT SOURCE</u>
2^0	Unavailable Execute External Instruction
2^1	Not assigned
2^2	Clock
2^3	Not assigned
2^4	Not assigned
2^5	Machine check
2^6	DMA 6
2^7	DMA 7

UNAVAILABLE EXECUTE EXTERNAL INSTRUCTION (bit 2^0) will be set and cause a class 3 interrupt when an EXECUTE EXTERNAL INSTRUCTION (EXI) is executed with an illegal or nonoperational sub-operation code.

CLOCK (bit 2^2) will be set and cause a class 3 interrupt when the delta clock is decremented to zero.

MACHINE CHECK (bit 2^5) will be set and cause a class 3 interrupt when a memory parity, I/O parity or Buffered DMA parity error condition is detected.

DMA 6 (bit 2^6) will be set and cause a class 3 interrupt when the device attached to buffered DMA channel 6 sets its interrupt line.

DMA 7 (bit 2^7) will be set and cause a class 3 interrupt when the device attached to buffered DMA channel 7 sets its interrupt line.

When a class 3 interrupt occurs it will force a jump to address 34, which will normally contain a GSB instruction to save P and generate a jump to the class 3 interrupt subroutine. The normal class 3 interrupt subroutine will first store the tally counter and store the designators and then execute an EXECUTE EXTERNAL INSTRUCTION (EXI) with a sub-operation code of 014_8 (SEE). The EXI instruction carries three operands after the sub-operation code with the interrupt data contained in the item specified by OP3. OP1 and OP2 of this instruction are not used but must have legal operands; it is a good policy to make OP1 and OP2 the same as OP3. A maximum of 14 bytes will be stored in OP3 with the following meaning:

1st byte of OP3 = Sub-Operation Code of Unexecutable Instruction

2nd byte of OP3 = Interrupt status (class 3 interrupt bits)

3rd byte of OP3 = OP1 lower address limits Bits 8-15

4th byte of OP3 = OP1 lower address limits Bits 0- 7

5th byte of OP3 = OP1 upper address limits Bits 8-15

6th byte of OP3 = OP1 upper address limits Bits 0- 7

7th byte of OP3 = OP2 lower address limits Bits 8-15

8th byte of OP3 = OP2 lower address limits Bits 0- 7

9th byte of OP3 = OP2 upper address limits Bits 8-15

10th byte of OP3 = OP2 upper address limits Bits 0- 7

11th byte of OP3 = OP3 lower address limits Bits 8-15

12th byte of OP3 = OP3 lower address limits Bits 0- 7

13th byte of OP3 = OP3 upper address limits bits 8-15

14th byte of OP3 = OP3 upper address limits Bits 0- 7

The 1st byte of OP3 will be the illegal or non-operational sub-operation code only if the Unavailable Execute External Instruction (bit 2^0) of the 2nd byte of OP3 is set. If no illegal or non-operational instruction was executed, the sub-OP code of 014_8 will be in byte 1. Byte 2 will contain the interrupt status while bytes 3 through 14 will carry the address limits of OP1, OP2 and OP3 of the EXI instruction.

In the case of the Machine check (bit 2^5 of byte 2) causing the interrupt the program must determine if a memory parity, I/O parity or Buffered DMA parity error caused the interrupt. The GOTO ON DESIGNATORS (GO) instruction can be used to

determine which parity error occurred. If a BDMA or I/O parity error occurred, the type of error and channel number can be determined by executing an EXI instruction with a sub-OP code of STORE CHANNEL PARITY ERROR (SCE). The octal code for this sub-op code is 015. (See EXI instruction format.) The Store Channel Parity Error sub-Op code will store one byte in the OP3 item which is bit position encoded as follows:

<u>BIT POSITION</u>		
2 ⁰	}	I/O channel on which I/O parity error occurred
2 ¹		
2 ²		
2 ³		Not assigned
2 ⁴	}	110 ₂ = BDMA Channel 6 parity error
2 ⁵		111 ₂ = BDMA Channel 7 parity error
2 ⁶	}	1 = BDMA address parity error
2 ⁷		0 = BDMA data or status parity error

NOTE: The above status is not valid unless an I/O or BDMA parity error occurred.

The Delta clock can be loaded to a 16 bit count by the EXI instruction using a sub-op code of LOAD DELTA CLOCK (LC) which has an octal code of 004. (See EXI instruction format.) The Delta clock is loaded with the first two bytes of OP1 item and will decrement at a 100 microsecond rate enabling time intervals from 100 microseconds to 6.5536 seconds. When the clock has decremented to zero a class 3 interrupt will occur with bit 2² of the interrupt status byte set. The Delta clock

can be program cleared by loading the clock with a value of binary zero. No interrupt will occur if the clock is program cleared.

DMA channel interrupts are dependent upon remote device being connected to DMA channels 6 and/or 7. The occurrence of these interrupts is dependent upon the condition of the remote device. Class 3 interrupts can be selectively enabled or disabled in OPl, byte 3 of the INTERRUPT MASK (IM) instruction.