ECE3073 Computer Systems

Practice Questions

Responding to events

i) Itemise and provide a brief explanation of the advantages of using interrupts to respond to external events.

1) Normal code can be written independently of the interrupt code. Unless you want the interrupt code and normal code to interact then you do not need to know anything about the interrupt code when writing the normal code.

2) It can be arranged that high priority interrupts can be serviced immediately with out waiting for lower priority tasks to be completed (as would be the case with polling).

3) If no interrupts occur than then no cpu time is used up by the interrupting process. With polling cpu time is used checking if service is required even when it is not.

ii) Itemise and provide a brief explanation of the disadvantages of using interrupts to respond to external events.

1) For interrupts the processor must execute a certain number of instructions during context switching. In some cases this context switching may take longer than a short polling loop.

2) The hardware of a processor will be more complex if it has to implement interrupt processing. This hardware must check for interrupts every instruction cycle and then perform the tasks required for context switching. There will also be additional registers to control interrupt processing and additional machinecode instructions. All of which translate to additional hardware.

iii) The following program provides service for two peripheral circuits (a and b) responding to external events.

```
loop:
        mov     r0,data_reg_a       ;read contents of status_reg_a    (1 T-state)
        and     r0,#0002h           ;test bit 1                       (1 T-state)
        jmpa    cc_NZ,service_a     ;if bit set jump to service routine (2/1 T-state/s)
        mov     r0,data_reg_b       ;read contents of status_reg_b    (1 T-state)
        and     r0,#0008h           ;test bit 3                       (1 T-state)
        jmpa    cc_NZ,service_b     ;if bit set jump to service routine (2/1 T-state/s)
        jmpa    cc_UC,loop          ;jump to start of polling loop    (2 T-states)
```

Note that the T-states given for each instruction are for the purposes of this question only. In a jump instruction the first number of T-states applies if the jump is taken, the second if execution continues straight on.

iiia) For peripheral circuit 'a' determine the maximum and minimum latency of this polling program in T states assuming that the computer is executing the polling loop when service is requested. Explain your reasoning behind the answers you give.

Maximum for peripheral 'a'

1 (mov - assume status bit in 'a' sampled at start of this instruction and just missed)
1 (and)
1 (jmpa)
1 (mov)
1 (and)
1 (jmpa)
2 (jmpa to loop)
1 (mov - status bit detected this time)
1 (and)
2 (jmpa – jump to service routine)

Total T-states for 'a' = 12

Minimum for peripheral 'a'

(1) (mov - assume status bit in 'a' sampled at end of this instruction and set just before. Therefore, do not count the time taken for this instruction)
1 (and)
2 (jmpa)

Total T-states for 'a' = 3

iiib)
For peripheral circuit 'b' determine the maximum and minimum latency of this polling program in T states assuming that the computer is executing the polling loop when service is requested. Explain your reasoning behind the answers you give.

Maximum for peripheral 'b'

1 (mov - assume status bit in 'b' sampled at start of this instruction and just missed)
1 (and)
1 (jmpa)
2 (jmpa to loop)
1 (mov)
1 (and)
1 (jmpa)
1 (mov - status bit detected)
1 (and)
2 (jmpa)

Total T-states for 'b' = 12

Minimum for peripheral 'b'

(1) (mov - assume status bit in 'b' sampled at the end of this instruction and set just before)
1 (and)
2 (jmpa)

Total T-states for 'b' = 3

iiic) Explain why the latency is the same for both peripheral circuit a and circuit b.

It can be seen that the maximum and minimum latencies are the same for 'a' and 'b' because the processor executes the same number of instructions in each case. For the maximum latency the order of instructions is slightly different but the total is still the same.

iv) Describe the circumstances in which synchronisation is necessary during data transfer.

If data is being transferred between two digital systems (perhaps a computer and a printer, or a computer and an analogue to digital converter) problems can arise due to the different speeds of the two systems. For instance, if a fast computer sends characters to a slow printer then the computer may send new characters before the previous ones have been printed and some later characters will be lost. If a fast computer reads data from a slow analogue to digital converter the computer may try to read new values from the ADC before they are available and read previous data instead. In both of these cases some method of synchronisation will cure the problem.

v) Explain how handshake signals can be used to synchronise data transfer from a fast computer to a relatively slow peripheral device.

In this case synchronisation is achieved by passing handshake signals between the computer and peripheral device as follows:

IDLE	Slow peripheral signals to the computer 'ready to receive data'
DATA READY	Computer places data on the data bus and then signals to the slow peripheral 'data now available'
DATA TAKEN	When the data has been received and dealt with by the slow peripheral it signals 'data has been taken' to the computer
IDLE	Slow peripheral indicates 'ready to receive data' again.

RAR 12/03/2012