3A3F Computer Architecture: Interfacing and Control

I D Reid
Bugs to ian.reid@eng
Updates on http://www.robots.ox.ac.uk/~ian/Teaching/ComputerArch/

Michaelmas 2002
Rev 24.9.2002

1. (a) Explain what is meant by the *address space* of a computer system.

   (b) Figure 1 shows a CPU with a 16-bit address bus to which are attached three different memory units, one ROM, one RAM and one memory area devoted to i/o. The diagram shows the range of address lines entering each device, and the address lines used to distinguish and select each unit.

   i. Deduce the maximum number of addressable registers in each memory device.

   ii. Deduce the part of address space occupied by each device, assuming that all undecoded lines are set to zero. (e.g.. for the ROM set A8-A12 to zero.)

   iii. You will have noticed that the circuit uses only partial decoding, and the devices are multiply mapped. How many times for the RAM? In (b)(ii) you worked out the address space range for the lowest mapping. What is the address space range for the next mapping up?

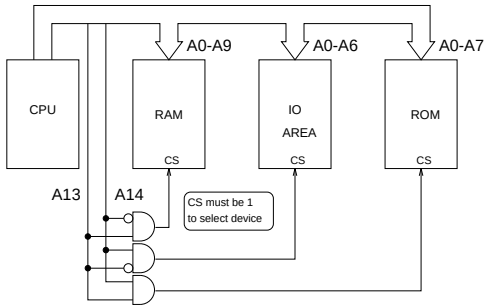


Figure 1: Memory and memory-mapped I/O devices

2. (a) What is meant by interrupt-driven I/O, and briefly discuss its advantages and disadvantages over other methods. Explain the terms maskable interrupt and vectored interrupt.

   (b) Three interrupt lines IRQ0, IRQ1 and IRQ2 each from a single peripheral enter a CPU which uses vectored interrupts. IRQ0, the highest priority interrupt, is non-maskable, but the progressively lower priority interrupts are masked so that an interrupt at the same or higher level can not itself be interrupted. The masks are three bits PSW[0], PSW[1], PSW[2] in the "process status word" PSW. PSW[0] is set high when handling an interrupt at level 0, and similarly for levels 1 and 2.

   The interrupt lines, which go high when requesting an interrupt, are tested at Location/Line 55 of the μPROM, at the start of a modified instruction-fetch cycle. If an interrupt is detected a jump is made to a section of RTL at Location/Line 80 which handles the interrupt.

   Write Line 55 in RTL to detect an interrupt and jump to the handler.

```
55.  fill this space
56.  MAR←PC
57.  MBR←M⟨MAR⟩
58.  IR←MBR;PC←PC+1
```

   (c) Your colleague has written the RTL microcode at line 80:

```
80.  M⟨SP⟩←PC; SP←SP−1
81.  M⟨SP⟩←PSW; SP←SP−1
82.  →(IRQ0,IRQ1,IRQ2)/(83,85,87)
83.  PC←M⟨VECTORBASE⟩
84.  PSW[0]←1;PSW[1]←0; PSW[2]←0;→(55)
85.  PC←M⟨VECTORBASE+1⟩
86.  PSW[1]←1; PSW[2]←0;→(55)
87.  PC←M⟨VECTORBASE+2⟩
88.  PSW[2]←1;→(55)
```

   Explain the operation and reasoning behind the code.

   (d) Somewhere you read that *non-maskable interrupts should be edge-driven rather than level-driven*, and you suddenly realize that the section of code from line 80 has a flaw. What is it, and what might be done about it?

   (e) Where should the interrupt services routines for each device be located? At the end of such a subroutine is placed the instruction RTI which differs from the usual RTS in one way. Write the register transfer microcode for RTI.

   (f) Explain briefly how a CPU might deal with several devices sharing the same interrupt line.

3. A CPU is connected to its memory and I/O peripherals using VMEbus. A 'talker' transfers data in parallel to a 'listener' on the bus using 8- or 16-bit words under the control of two handshake lines, DS* (Data Strobe) and DTACK* (Data Acknowledge), where the * indicates that the lines are active when 0, or 'active low'.

   The RTL description of a listener for one mode of data transfer is:

```
MODULE: LISTENER
MEMORY: VALUE[8], ackff
INPUTS: DATA[8], ds
OUTPUTS: dtack
    1 ackff ←0
    2 →(ds)/2
    3 ackff ←1; VALUE ←DATA
    4 →(ds,d̄s̄)/(1,4)
ENDSEQUENCE
CONTROLRESET(1); dtack=ackff
END
```

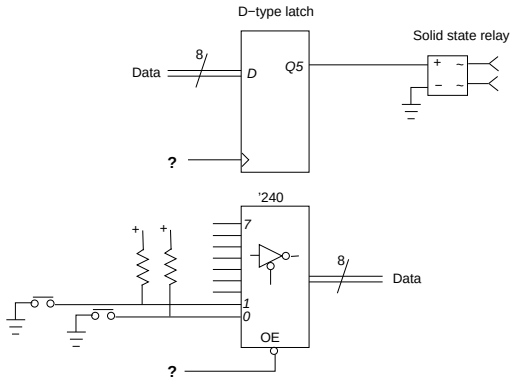


Figure 2: A circuit for digital I/O

(a) Draw a timing diagram for the listener's signals, starting at line 1 of the RTL and showing ackff, ds, data, and dtack. Annotate the diagram so that the handshaking process is clear. You should assume that the delay between talker and listener of 1μs **and vice versa** is much greater than a clock period.

(b) Estimate the maximum throughput in bytes per second.

(c) Write the RTL description for the talker. (Hint. The first think the listener says is "ready to listen". So you should arrange for the talker to wait for this. Also, as the next thing the listener does is wait for ds to go low, the talker better not have ds low to start with.)

(d) Is this communication synchronous or asynchronous? What are the pros and cons of an asynchronous bus versus a synchronous one?

4. Consider the circuit show in Figure 2, consisting of a digital input and a digital output (note that the '240 chip is an inverting buffer).

   It is to be interfaced to a system which has a memory address space of 64K and a port address space of 1K, and which has bus control lines which include IOW*, IOR*, MEMW* and MEMR* (all all active *low*).

   (a) Identify which part of the circuit corresponds to output and which to input

   (b) If the circuit is to be fully decoded, how many address lines must be decoded if the device is (i) memory mapped (ii) port mapped

   (c) Design a decoding circuit using an 8-bit binary comparator (i.e. a chip with 2 sets of 8 inputs and whose output is high only if the two sets of 8 are equal) and whatever miscellaneous gates (AND, NAND, etc) you need, to decode *both* input and output ports so that they are *port-mapped* to address 1FA (hex). How can we have two devices at the same address?

(d) The solid-state relay controls a pump and the pair of switches are mounted vertically above one another to indicate the level of fluid in a tank. A switch is closed (on) if it is uncovered. If the lower switch is uncovered the pump should be turned on, but the pump should be turned off if the upper switch is covered (open/off). Write short polling loop of (pseudo) assembly language instructions to achieve this.

(e) Consider what modifications would be required to the circuit and to your code if the circuit were to be memory mapped

(f) If the fluid is the coolant in a water-cooled nuclear reactor, a low level probably indicates a dangerous situation developing. Briefly discuss the inherent dangers of a software only solution in this situation.

5. A micro-controller is responsible for the local control of a small 4-axis robot, and runs a variety of simple tasks via a simple pre-emptive priority scheduler.

   In particular, one process "`SERVO`" runs a PID servo-loop controlling all four axes in both position and velocity, at 2ms intervals. A second process "`SERIAL`" monitors the micro-controller's RS232 serial port, on which new demands periodically appear.

   (a) You are told that the RS232 port operates at 9600 baud. What does this mean?

   (b) Clearly by storing the demands in a shared data area the `SERIAL` process can communicate them to `SERVO`. Why would it be disastrous if `SERIAL` wrote the data into this area *as it arrived*? A better idea is for `SERIAL` to wait until a complete set of demands has arrived, and then write them to the shared data area. Care must still be exercised, but why?

   (c) Define a *semaphore*, and describe how one can be used to solve the problem above

   (d) An alternative solution would be to use synchronous *channel* communication. Describe (with pseudo-code if you wish) how this would be accomplished. Would you use blocking or non-blocking I/O for (i) `SERVO`, (ii) `SERIAL`

   (e) Define *deadlock* and describe an example.

6. (a) For a sampled data system, give reasons why increasing the sample rate (if possible) is beneficial.

   (b) Using the property that the transfer function of a system is equal to its impulse response, derive the Laplace Transform of a Zero-Order Hold filter with hold time Tseconds.

   (c) Determine the bandwidth of the ZOH filter

   (d) For frequencies within this bandwidth, what is the maximum phase lag of the ZOH?

   (e) In digital control applications, why is ZOH used rather than something which better approximates an ideal low-pass filter? When might a sharp cut-off filter be appropriate?

7. A PID controller with $K = 5, T_d = 0.0008$ and $T_i = 0.003$ is to be used to control a micro-servo motor with transfer function from input voltage to output speed (rad/sec) of

$$G(s) = \frac{360000}{(s+60)(s+600)}$$

   A tacho-generator produces a voltage proportional to the speed (but for the remainder of this exercise assume that the constant of proportionality is one; i.e. we have unity feedback)

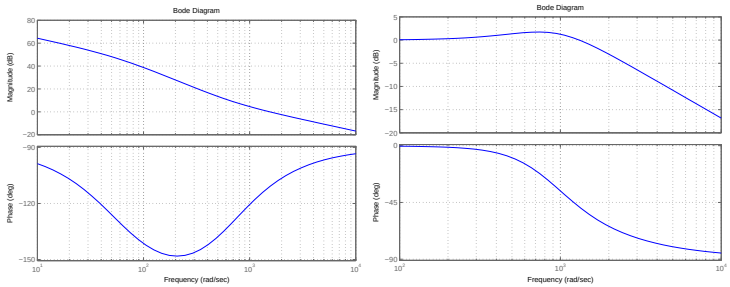
(a) Draw a block diagram of the continuous system

(b) Discuss the hardware aspects of converting this system into a digitally controlled one and draw an annotated block diagram of the digital system

(c) Using forward differences (and both forward and backward differences for 2nd derivatives), derive a difference equation to replace the continuous PID controller

(d) Figure 3 shows bode plots for (a) the open-loop and (b) closed-loop systems. Hence (or otherwise) estimate the phase margin and the closed-loop bandwidth.

(e) For a sample rate of 10 times the bandwidth, find an approximate value for the phase margin of the system. Repeat for sampling frequencies of 5, 20, 30 and 40 times the closed-loop bandwidth. Hence estimate the system damping in each case and plot this against sampling frequency.



(a) (b)

Figure 3: Bode plots for (a) open-loop and (b) closed-loop systems