EEE336 Examination Solutions – 2015/2016

Question 1

(a)

(i) Implied mode

The operand is specified implicitly in the definition of the opcode. An example of this is the Intel 8085 command CMA which complements the accumulator. **[2 marks]**

(ii) Immediate mode

The operand is specified in the instruction itself and is thus immediately available for use. Useful for initializing registers to a constant value. **[2 marks]**

(iii) Register Indirect

Operand is held in a memory cell pointed to by the contents of the register. i.e. the register in the processor contains the address of the operand in memory. **[2 marks]**

(iv) Based Memory Addressing

The instruction contains a pointer to the base address and a displacement. The effective address is formed by adding the displacement to the base address. **[2 marks]**

(b)

Sign extending each operand to 8 bits gives: 00001101 ÷ 00000100

Initially, we form 00001101 – (00000100 x 2^3) = 00001101 – 00100000, which can be most easily achieved by taking the 2s-complement of 00100000 = 11100000.

Therefore**, 00001101 – 00100000 = 00001101 + 11100000 = 11101101**

(As a check, = -19). Therefore, $q_3 = 0$

The second stage is: **11101101 + (00000100 x 2^2) = 11101101 + 00010000 = 11111101**

(As a check, -3). Therefore, $q_2 = 0$.

The third stage is: **11111101 + (00000100 x 2^1) = 11111101 + 00001000 = 00000101**

(As a check, +5). Therefore, $q_1 = 1$.

The final stage is: 00000101 – (00000100 x 2^0) = 00000101 – 00000100. Taking the 2s-complement of 00000100 gives: 11111100. Thus:

00000101 – 00000100 = 00000101 + 11111100 = 00000001

Since this is positive, q1 = 1 and the remainder is 1

Therefore the answer is $q_3q_2q_1q_0$ = 0011 remainder 00000001. (13 ÷ 4 = 3R1)

Mark Allocation:

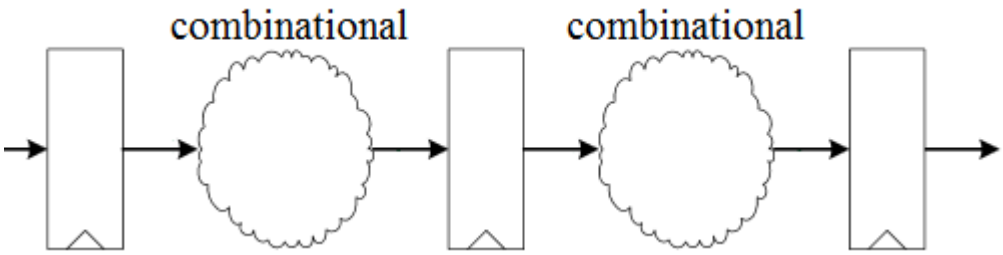
2 marks for setting up the problem as stated in the question

3 marks for correct calculations at each stage using 2s complement

2 marks for correct non-restoring method and extraction of correct result

(c)

Synchronous Pipeline:



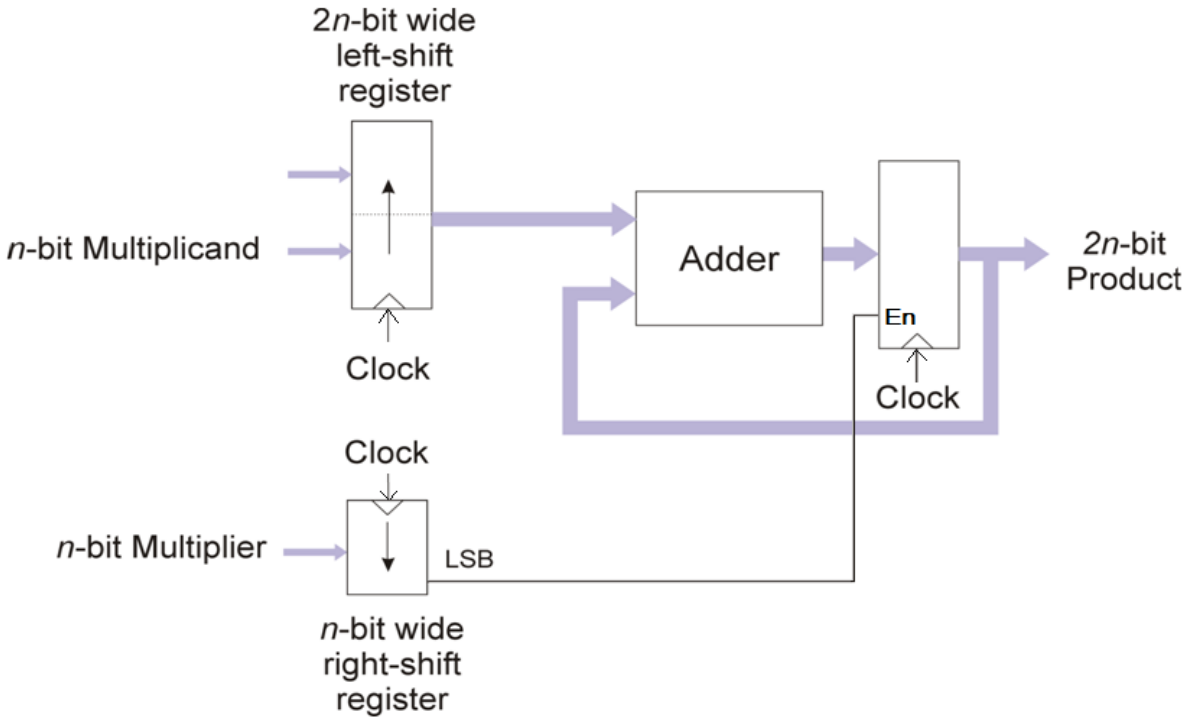
Combinational circuitry is divided between synchronous register stages to improve performance. **[2 marks]**

A non-blocking Verilog coding style would be preferred as the register assignment statements are deferred and take place at the end of the simulation step. The correct functionality will be obtained irrespective of the statement ordering. With blocking assignments, evaluation and assignment are immediate and hence the functionality obtained is dependent on the statement ordering. **[3 marks]**

Question 2

(a)

Sketch a possible hardware implementation of a serial shift-and-add multiplier (using a parallel adder) for multiplying two n-bit numbers. Carefully describe the initial circuit conditions.



i) The multiplicand is loaded into the bottom *n* bits of the 2*n*-bit wide multiplicand shift register. (The upper *n* bits are zero.)

ii) The multiplier is loaded into the multiplier shift register.

iii) The 2n-bit wide accumulator register is loaded with zero. (This register holds the running total of the partial products.)

[6 marks]

For the more general case of multiplying an m-bit multiplicand by an n-bit multiplier, how long will it take to generate the product?

As one partial product has to be accumulated for each bit of the multiplier, the product will require n clock cycles to generate. **[2 marks]**

Outline the operation of a serial shift-and-add multiplier with reference to the multiplication of two unsigned integers, 110_2 by 101_2 using a table to show the states of the circuit at each stage.

Circuit state immediately before each clock edge:

Multiplicand Register	Multiplier Register	Output of Adder	Product Register
000110	101	000110	000000
001100	*10	010010	000110
011000	**1	00111100	000110
Don't care	Don't care	Don't care	011110

After 3 clock cycles the answer(= 011110) is available in the product register. **[4 marks]**

(b)

Using the example of 110_2 multiplied by 011_2 (where 110_2 is the multiplicand and 011_2 is the multiplier) describe how basic shift and add multiplication can be extended to handle multiplication of a signed multiplicand by an unsigned multiplier.

Unless it is adapted, basic shift-and-add multiplication will give the wrong answer for a signed multiplicand. In particular, for the product of two n-bit numbers, the product is 2n bits wide and the multiplicand has to be sign extended to 2n bits in order to correctly preserve the sign of the product. Thus:

```
111110      NB - Sign extended multiplicand
   011  ×
------
111110
111100
000000
------
111010
------
```

[4 marks]

(c)

In terms of hardware implementation, compared to the serial multiplier circuit in (a), it is possible to replace the 2n-bit wide multiplicand register with just an n-bit wide register. The principal advantage, however, can be seen by inspection from the table describing operation: namely, we only require an n-bit wide adder instead of the 2n-bit wide adder required in (a). This produces a saving in hardware which can be significant if n is large.

[4 marks]

Question 3

(a)

(i)Why do RISC machines contain a large number of registers?

This is to reduce the number of accesses to main memory which can take several cycles. In comparison to main memory, access time to a register is negligible. Thus, programs should run faster. **[2 marks]**

(ii)What is the principal advantage gained by reducing the number of instructions available in a RISC machine?

This leads to simpler logic for decoding and execution, freeing up space for more registers.

[2 marks]

(iii)Why is the instruction set in a RISC machine limited to simple instructions?

The simpler instructions can be executed in a single cycle. This makes the implementation of multi-stage pipelines easier.

[2 marks]

(b)

(i)The types **wire** and **reg**

wire – net data type used for connections, does not store any data, must be driven.
reg – variable data type used as a storage element, stores value between assignments, holds its value until a new assignment is made. **[2 marks]**

(ii)An **initial** procedure and an **always** procedure

initial procedure – executes once only, never repeats, single-pass behaviour
always procedure – cyclic behaviour, executes whenever there is an event on its sensitivity list **[2 marks]**

(iii) Inertial and **transport** delay

Inertial Delay is a delay model which indicates the amount of time that a signal change must persist at the input of a logic gate in order for a change to propagate to the output. A pulse which does not last longer than the inertial delay will be suppressed and there will not be any change at the output. Pulses of length greater than the inertial delay will propagate to the output after that delay.

Transport delay does not have this restriction and changes on the input will be reflected at the output after the specified delay. **[3 marks]**

(c)

Truth table for Verilog code :

a1	a0	y0	y1	y2	y3
0	0	1	0	0	0
1	0	0	1	0	0
0	1	0	0	1	0
1	1	0	0	0	1

[3 marks]

Function:

2-to-4 line decoder **[1 mark]**

New code:

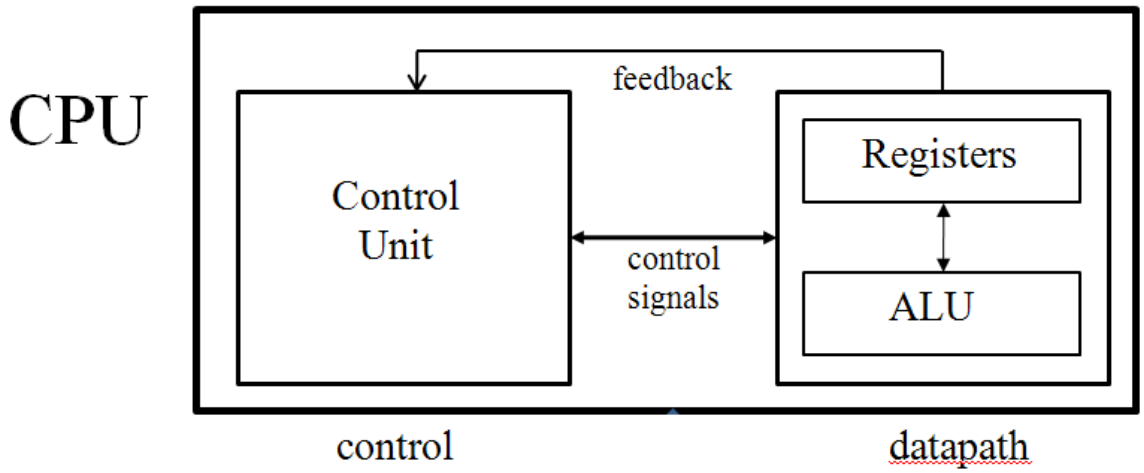
```
module digcomponent (output reg [7:0] y, input [2:0] a);
integer i;
always @ (*)
  for (i = 0; i <= 7; i = i + 1)
    if  (a == i)
      y[i] = 1;
    else
      y[i] = 0;
endmodule
```

[3 marks]

Question 4

(a)

What are the two principal elements that make up a Central Processing Unit (CPU) and how are they connected. Describe briefly.



The CPU consists of a datapath and control logic. The datapath performs arithmetic and logical operations using registers as temporary storage. The control unit determines the sequence of these operations by decoding and executing instructions from a program. Some status information is fed back to the control path and can be used to make decisions on the program flow.

[4 marks]

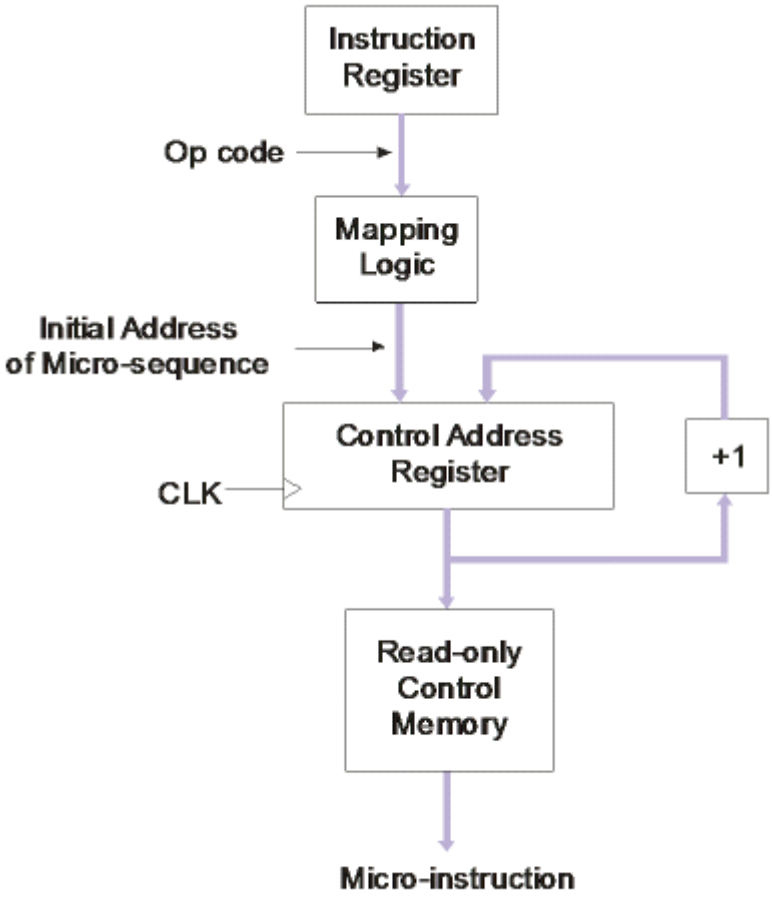
(b)

In a central processing unit (CPU), the sequence of control signals can be generated either using hardwired logic or a look-up table (microcoding). What is the advantage of the microcoding approach?

It is easier to design and modify, particularly for complex control paths. **[2 marks]**

(c)

Sketch a simple microcoding arrangement, briefly describing the operation of each functional block.



[4 marks]

i) The Instruction Register holds the op code of the machine instruction which is to be translated into a sequence of Micro-instructions.
ii) The Mapping Logic determines the address in the Control Memory of the start of the sequence of Micro-instructions.
iii) The Control Address Register holds the address of the current Micro-instruction. The value in this register is incremented by the system clock to advance through the sequence of Micro-instructions.
iv) The Control Memory is a read-only memory, the contents of which are fixed at point of manufacture. The Control Memory holds the sequence of Micro-instructions corresponding to a given op code, stored in ascending locations.

[6 marks]

(d)

What is the disadvantage of generating micro-instructions using the simple microcoding arrangement in **(c)** above? Describe how *vertical microcoding* can be used to address this problem. What is the disadvantage of using *vertical microcoding*?

The principal disadvantage of the direct microcoding scheme described in (c) is that the size of the Control Memory can become very large, particularly for complex processors. Vertical microcoding implements what are effectively compression techniques which, instead of directly encoding the micro-instructions, hold the information in a form which can be recovered by a look-up table. The disadvantage of this is that the additional processing steps can limit the overall clock speed of the processor.

[4 marks]