

IMPERIAL COLLEGE OF SCIENCE TECHNOLOGY AND MEDICINE
UNIVERSITY OF LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
M.Eng., B.Eng., B.Sc(Eng.) and A.C.G.I. EXAMINATIONS 2001

PART III and PART IV

DIGITAL SYSTEM DESIGN

There are **SEVEN** Questions.

Answer **FOUR** questions.

First Marker: *PYKC*

Second Marker: *DMB*

Special instructions for invigilators: None

Information for candidates:

In the figures showing digital circuits, all components have, unless explicitly indicated otherwise, been drawn with their inputs on the left and their outputs on the right. All signals labelled with the same name are connected together. All circuits use positive logic. The least significant bit of a bus signal is labelled as bit 0, and the most significant bit with the highest integer number. Therefore the signal X7:0 is an eight bit bus with X7 being the MSB and X0 the LSB.

Hexadecimal numbers are prefixed with '\$'. For example the decimal number 10 is written as \$A.

In questions involving circuit design, you may use any standard digital circuits that are not explicitly forbidden by the question provided that you fully specify their operation.

Marks may be deducted for unnecessarily complex designs unless you are explicitly instructed not to simplify your solution.

1. The finite state machine shown in *Figure 1a* detects 4-bit sequences on a binary input X and produces a high pulse for one clock cycle at the output Z if either 0110 or 1010 pattern occurs. The timing of the state machine is illustrated in *Figure 1b*. The input X changes shortly after the rising edge of the clock. Detection starts on the clock edge after RESET is deasserted and the input X is processed by the FSM as groups of 4 bits. The output Z is asserted during the fourth bit of the input pattern.

a) Draw a state diagram and a state transition table for the FSM.

[12 marks]

b) Use either state merging method or implication table, reduce this state machine to 8 states or fewer. Draw the reduced state diagram and its equivalent state transition table.

[8 marks]

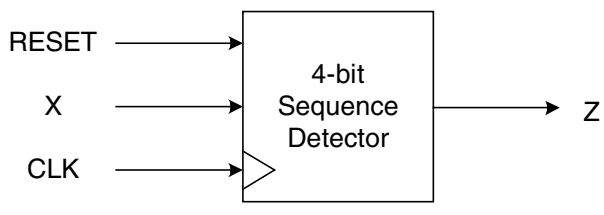


Figure 1a

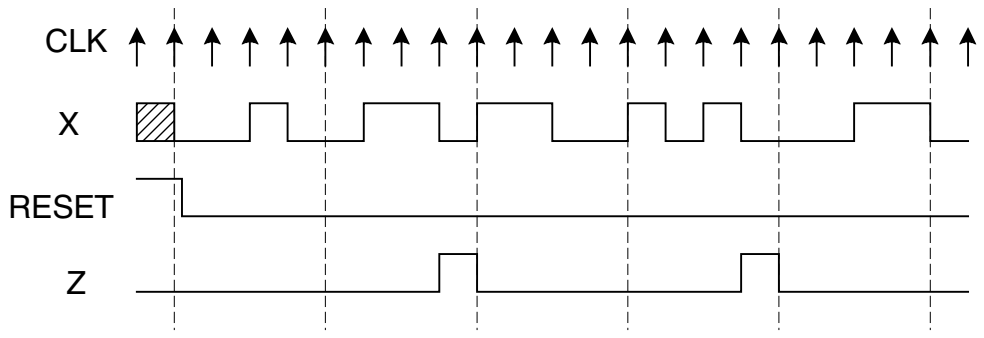


Figure 1b

2. An Altera's Flex10K series field programmable gate array (FPGAs) contains Embedded Array Blocks (EABs) that can be configured as a 256 x 8 ROM. The content of the ROM is given by the equation $255 * \sin(\pi * A / 512)$ rounded to the nearest integer, where A is the address of the ROM from 0 to 255.

Design a circuit using this EAB and other non-EAB logic cells on the FPGA to generate an output Y in 2's complement form, such that:

$$Y = 255 * \sin(\pi * N / 512)$$

where N is an input to the circuit representing an unsigned integer in the range of 0 to 1023 and Y is rounded to the nearest integer. State clearly the width of the datapaths in your design.

Explain the factors that limit the operating system clock frequency for your circuit and show how the frequency can be maximized.

Demonstrate that your circuit gives the correct answer for inputs of \$300 and \$380.

[20 marks]

3. *Figure 2a* shows the block diagram of a circuit to compute the square root R of an 8-bit unsigned integer P . The digital comparator (M1) compares the two unsigned inputs A and B , and produces an output $AGEB$ if $A \geq B$. The squaring circuit (M2) multiplies its 4-bit input by itself to produce an 8-bit square value S .

The successive approximation controller (SAC) performs the successive approximation algorithm to find the square root of P . It is initiated by a one cycle pulse on $START$ synchronised to the positive edge of CLK . It then successively asserts one bit per clock cycle from the most significant bit to the least significant bit. The asserted bit is kept if $AGEB = 1^1$ (i.e. $P < R^2$), and reset to zero otherwise. After four clock cycles, R contains the largest integer such that $R^2 \leq P$. The timing of the SAC circuit is illustrated in *Figure 2b* for $P=101$.

Design a circuit to implement the function of the SAC. Your design may be a finite state machine or a combination of finite state machine and other circuits. You only need to specify the finite state machine in the form of a state diagram or ASM chart.

[20 marks]

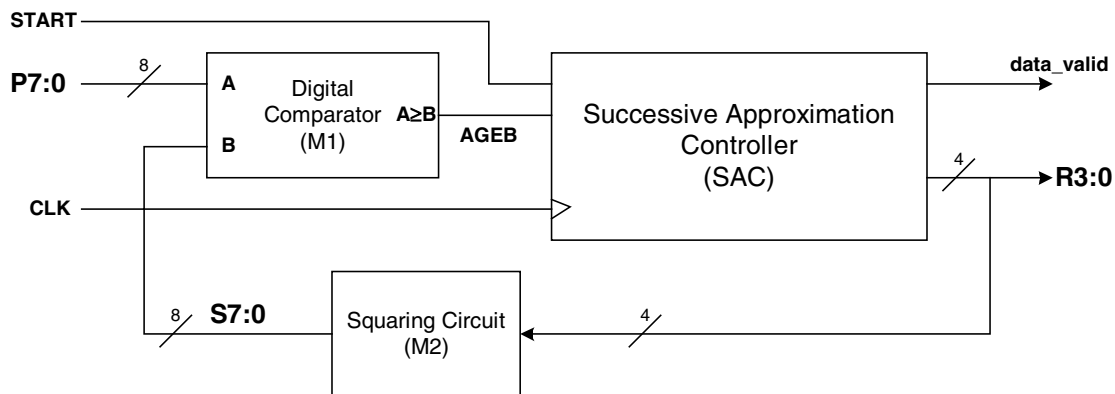


Figure 2a

a) _____

¹ Original stated $AGEB=0$, which is wrong. This was not raised during the examination, but is corrected here.

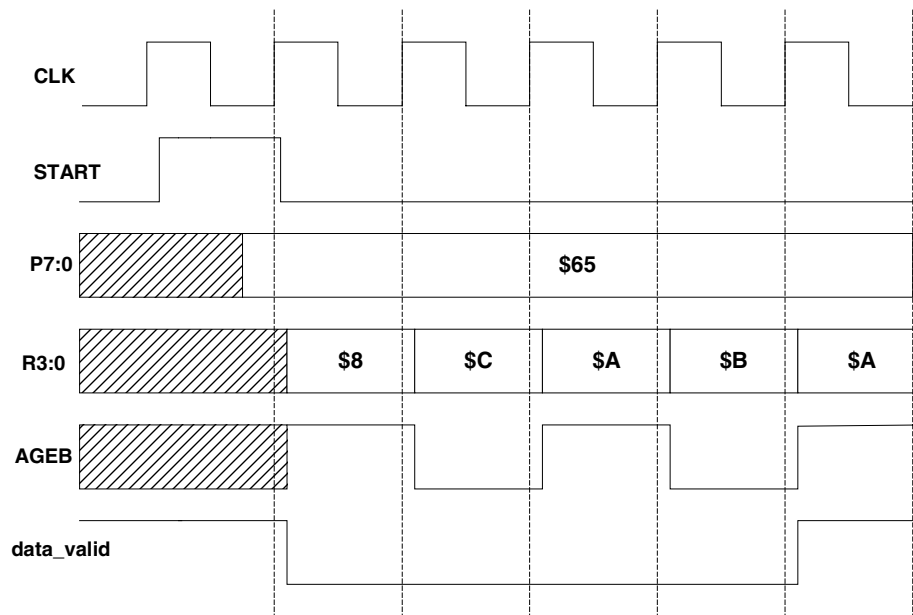


Figure 2b

4. Messages composed of five symbols from the set (A, B, C, D, E) are to be transmitted over a serial link. The variable length Huffman code is used to encode the data stream. The binary representation and the frequency of occurrence for the five symbols are:

| Symbol | Binary Code | Probability of Occurrence |
|--------|-------------|---------------------------|
| A | 000 | 0.25 |
| B | 001 | 0.2 |
| C | 010 | 0.2 |
| D | 011 | 0.19 |
| E | 100 | 0.16 |

- (a) Design a suitable set of Huffman code for this set of symbols.

[4 marks]

- (b) The continuous stream of received data is translated from Huffman code back to binary code with a finite-state machine decoder as shown in *figure 3*. All signals change on the rising edge of CLK. DIN is the serial data stream containing the message in Huffman code designed in (a). DOUT is the binary code output. It should contain the valid binary code during the clock cycle in which the last bit of each transmitted symbol is received. During this cycle, the VALID signal is high. Draw the state diagram of the finite state machine assuming that the state machine is initialised to state 0 on power-up and always return to state 0 at the beginning of each symbol.

[6 marks]

- (c) Using one-hot encoding, design the logic circuits required to implement this finite state machine in Boolean equation form. It is not necessary to minimise the logic equations nor obtain the optimal state assignment.

[8 marks]

- (d) Assuming the finite state machine is implemented with an Altera Flex10K FPGA with 4-LUT based logic cells, estimate with justification the number of logic cells required to implement your design.

[2 marks]

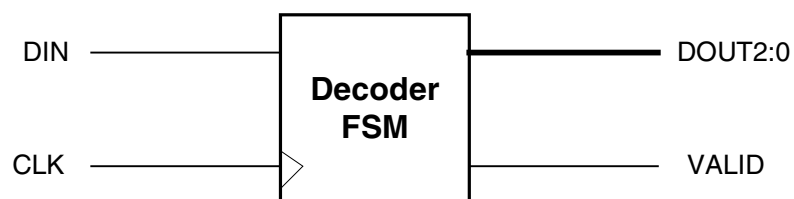


Figure 3

5. *Figure 4a* shows a systolic processing cell consisting of a coefficient register M1, a 2's complement multiplier M2, an adder M3 and two D-type registers M4, M5. The coefficient register is preloaded with a constant value a_i .

Figure 4b shows two such cells connected together to form a linear array of 2 cells. An input sample X is supplied on each clock cycle to the P_{in} input of the left-most cell. The array output Y and Z are taken from the A_{out} and P_{out} outputs of the right-most cell respectively. a_2 is a constant value supplied to the A_{in} input of the left-most cell.

a) Assuming that at time t , the input data to the array is x_n , deduce the relationship between the outputs Y and Z of the array in terms of x_n two clock cycles later. Draw a timing diagram to illustrating your answer.

[12 marks]

b) Hence or otherwise, state a general relationship between Y and X for a linear array of N cells.

[5 marks]

c) Assuming that X comes from a D-type register and that the delays of the circuit elements inside the systolic processing cells are: multipliers 25ns, adder 10ns, register 10ns, registers setup time 5ns, register hold time 0ns. What is the maximum operating frequency of such a linear array with N cells?

[3 marks]

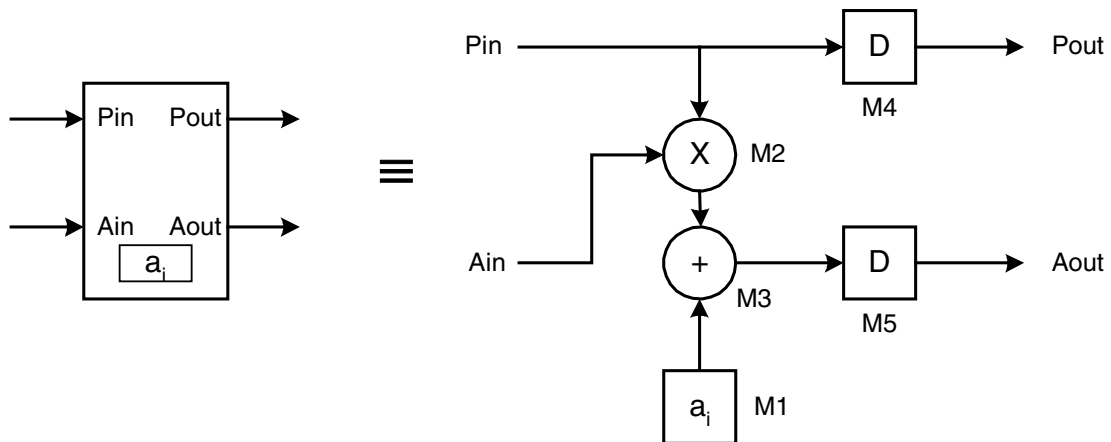
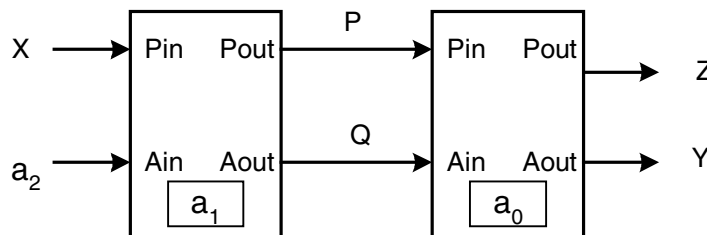


Figure 4a

Figure 4b



6. *Figure 5* shows an error-correcting memory module that can correct single bit error in the data D7:0 using Hamming codes. The signal ERROR at the output is high when an error is detected and low otherwise.

a) Design the parity generator circuit in the form of Boolean equations.

[6 marks]

b) Design the parity checking circuit and the error correction circuit. Demonstrate that your design works properly with the example D7:0 = 01101001 and an error has occur in D0. Gate level design is not required.

[14 marks]

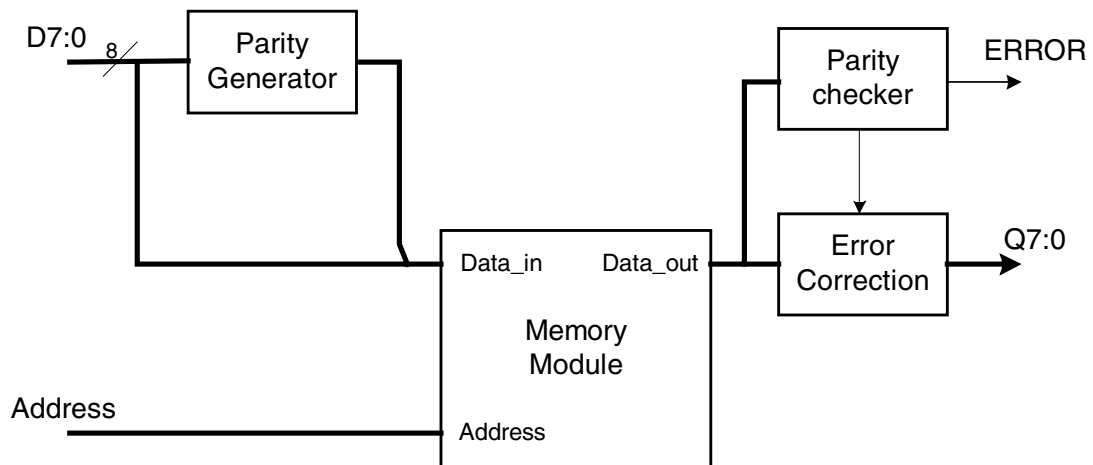


Figure 5

7. Figure 6 shows a microprocessor connected to a 4M x 32 bit SDRAM module through an address decoder and an interface circuit. The timing of the microprocessor and the SDRAM for reading a burst of four memory words is shown in Figure 6. The address bus $A_{31:0}$ becomes valid and the address strobe signal \overline{AS} is asserted at time B after the rising edge of the clock signal CLK during the clock cycle T_0 . The period of the clock is A. The address decoder circuit produces a chip select signal \overline{CS} , which is asserted low for addresses in the range \$00000000 to \$003FFFFFF provided that \overline{AS} is also low at time C after \overline{AS} is asserted. The row address strobe signal \overline{RAS} and the column address strobe signal \overline{CAS} are asserted by the SDRAM interface circuit for one clock cycle on the falling edge of the clock during T_1 and T_5 respectively.

The microprocessor reads four consecutive words from SDRAM on the rising edge of the clock if the \overline{DTACK} signal is sampled low.

Design the address decoder and the SDRAM interface circuit to generate \overline{CS} , \overline{RAS} , \overline{CAS} , \overline{DTACK} and the memory address bus signals $AD_{10:0}$. State the relationship between the time periods A, B and C, and any other constraints under which your design is valid.

[20 marks]

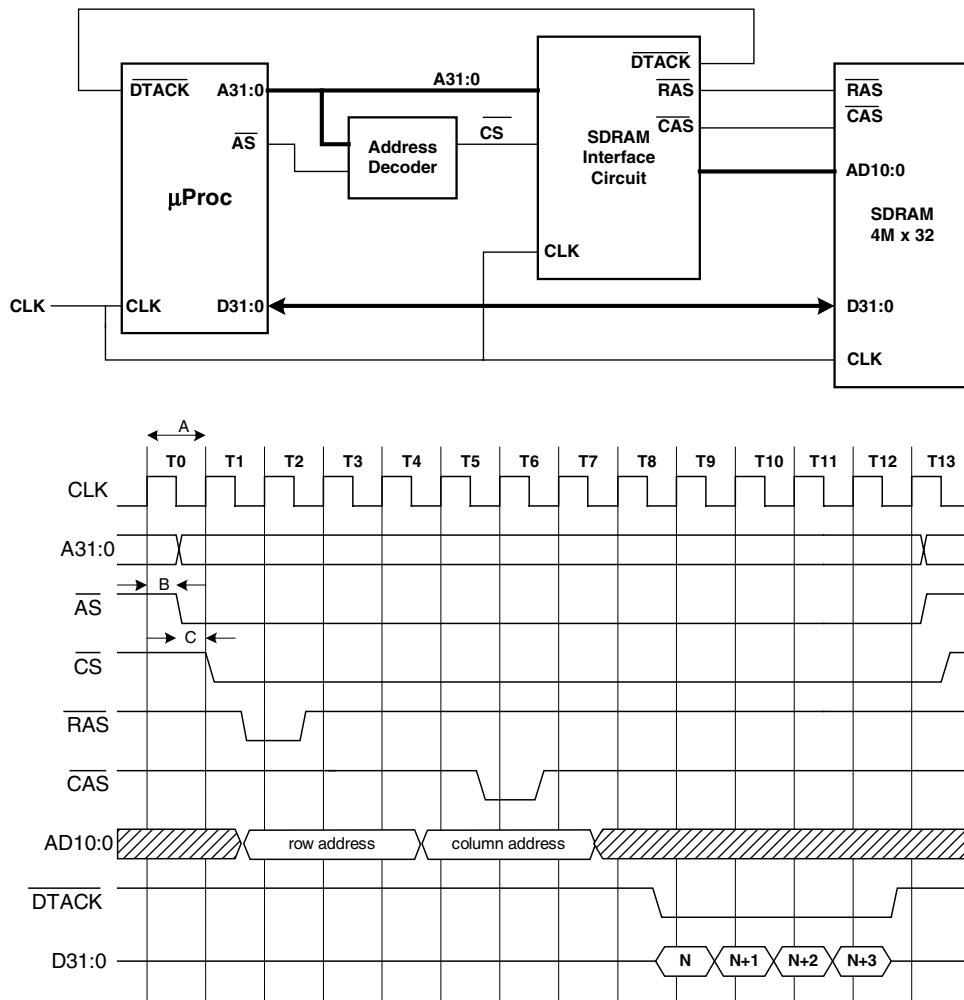


Figure 6

[END]

IMPERIAL COLLEGE OF SCIENCE TECHNOLOGY AND MEDICINE
UNIVERSITY OF LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
M.Eng., B.Eng., B.Sc(Eng.) and A.C.G.I. EXAMINATIONS 2001

PART III and PART IV

DIGITAL SYSTEM DESIGN

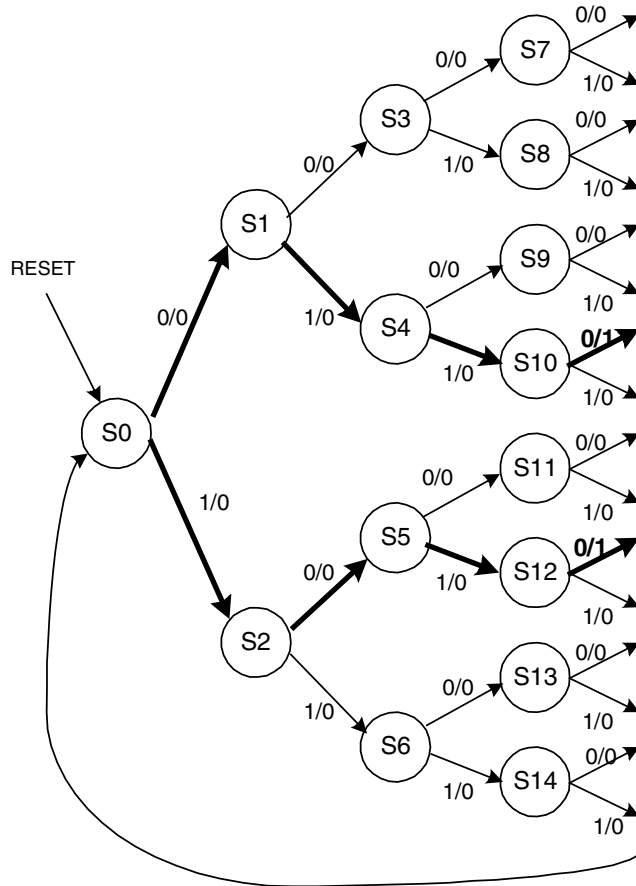
SOLUTIONS

First Marker: *PYKC*

Second Marker: *DMB*

Answer to Question 1

(a)



Upper bound 15 states.

[12 marks]

(b)

| Current State | Next State | | Output Z | | This column not required |
|---------------|------------|-----------|----------|----------|--------------------------|
| | X=0 | X=1 | X=0 | X=1 | Sequence detected |
| S0 | S1 | S2 | 0 | 0 | None so far |
| S1 | S3 | S4 | 0 | 0 | 0 |
| S2 | S5 | S6 | 0 | 0 | 1 |
| S3 | S7 | S8 | 0 | 0 | 00 |
| S4 | S9 | S10 | 0 | 0 | 01 |
| S5 | S11 | S12 | 0 | 0 | 10 |
| S6 | S13 | S14 | 0 | 0 | 11 |
| S7 | S0 | S0 | 0 | 0 | 000 |
| S8 | S0 | S0 | 0 | 0 | 001 |
| S9 | S0 | S0 | 0 | 0 | 010 |
| S10 | S0 | S0 | 1 | 0 | 011 |

| | | | | | |
|------------|-----------|-----------|----------|----------|------------|
| S11 | S0 | S0 | 0 | 0 | 100 |
| S12 | S0 | S0 | 1 | 0 | 101 |
| S13 | S0 | S0 | 0 | 0 | 110 |
| S14 | S0 | S0 | 0 | 0 | 111 |

b) Apply state merging (two states are equivalent if next states and outputs are the same):

S7≡S8≡S9≡S11≡S13≡S14 merged to SA

S10≡S12 merged to SB

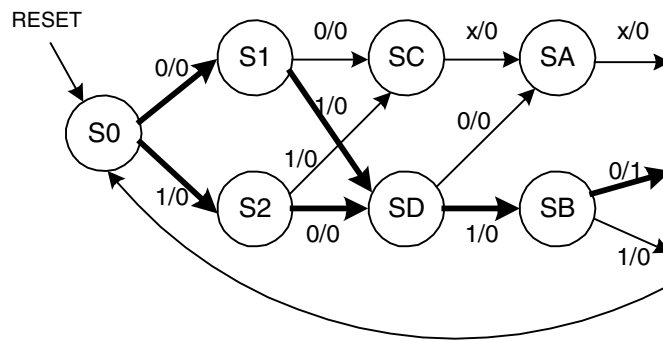
| Current State | Next State | | Output Z | | Sequence detected |
|---------------|------------|-----------|----------|----------|-----------------------|
| | X=0 | X=1 | X=0 | X=1 | |
| S0 | S1 | S2 | 0 | 0 | None so far |
| S1 | S3 | S4 | 0 | 0 | 0 |
| S2 | S5 | S6 | 0 | 0 | 1 |
| S3 | SA | SA | 0 | 0 | 00 |
| S4 | SA | SB | 0 | 0 | 01 |
| S5 | SA | SB | 0 | 0 | 10 |
| S6 | SA | SA | 0 | 0 | 11 |
| SA | S0 | S0 | 0 | 0 | xxx except 011 or 101 |
| SB | S0 | S0 | 1 | 0 | 011 or 101 |

S3≡S6 merged to SC

S4≡S5 merged to SD

Final reduced state transition table:

| Current State | Next State | | Output Z | | Sequence detected |
|---------------|------------|-----|----------|-----|-----------------------|
| | X=0 | X=1 | X=0 | X=1 | |
| S0 | S1 | S2 | 0 | 0 | None so far |
| S1 | SA | SD | 0 | 0 | 0 |
| S2 | SD | SC | 0 | 0 | 1 |
| SC | SA | SA | 0 | 0 | 00 or 11 |
| SD | SA | SB | 0 | 0 | 01 or 10 |
| SA | S0 | S0 | 0 | 0 | xxx except 011 or 101 |
| SB | S0 | S0 | 1 | 0 | 011 or 101 |



[8 marks]

Note: Some students may start with the optimised state diagram in part a), in which case, part b) should show that it cannot be reduced further.

Answer to Question 2

The EAB in 256 x 8 bit configuration stores 1/4 cycle of a sine table such that:

$$\text{Mem}[K] = 255 * \sin(\pi * K / 512) \text{ for } K = 0 \text{ to } 255.$$

[2 marks]

Generate the other quadrants by manipulating the address and negating the ROM/RAM values.

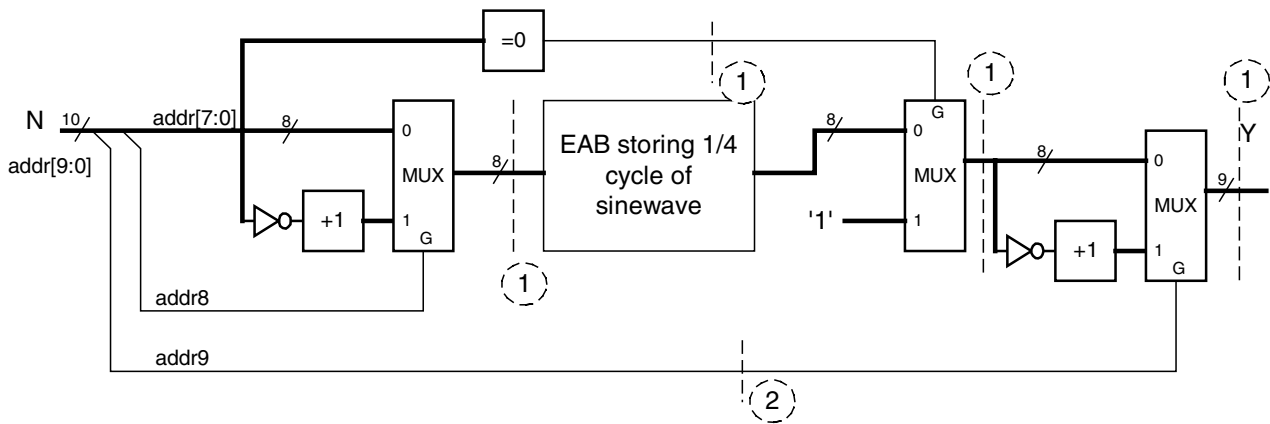
The rule to generate the EAB address 'reflection' and amplitude negation are:-

| addr9 | addr8 | Address to EAB | Negation |
|-------|-------|-----------------|----------|
| 0 | 0 | addr[7:0] | No |
| 0 | 1 | 512 – addr[7:0] | No |
| 1 | 0 | addr[7:0] | Yes |
| 1 | 1 | 512 – addr[7:0] | Yes |

[4 marks]

This works except for N=256 and 768 when addr[7:0] = 0. Therefore, detect this condition and force output to either +255 or -255.

[2 marks]



[6 marks]

To maximize the clock frequency that this module can work under, insert pipelines at vertical lines as shown. Shown in circles are the numbers of pipeline stages required.

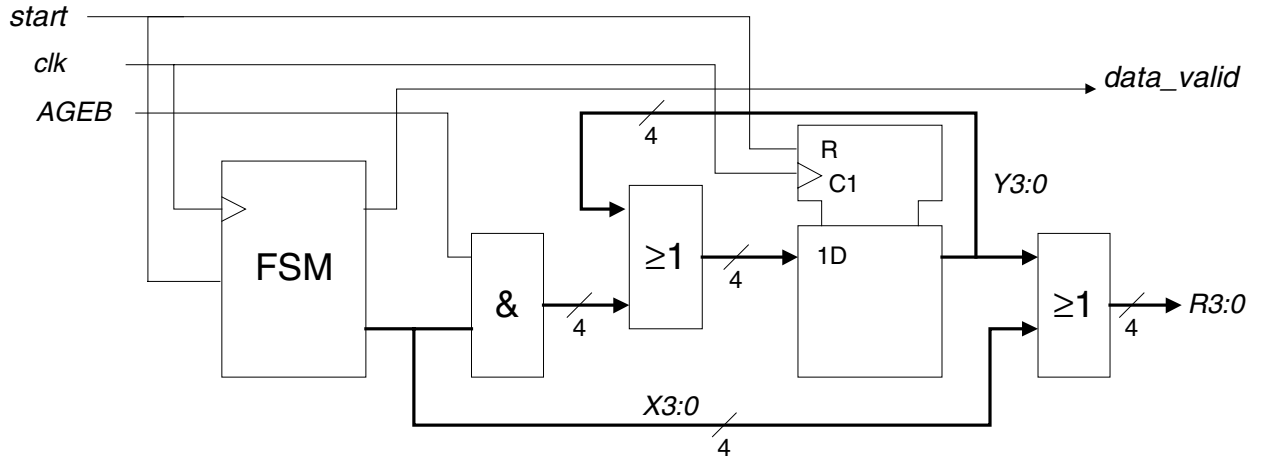
When N = 768, EAB output not selected. Addr9 = 1, therefore output jammed to -255.

When N = 896, addr[7:0] = 1000 0000, addr8 = addr9 = 1. Therefore, address supplied to EAB memory is 128. The content at this location is 180. This is then negated to give Y = -180.

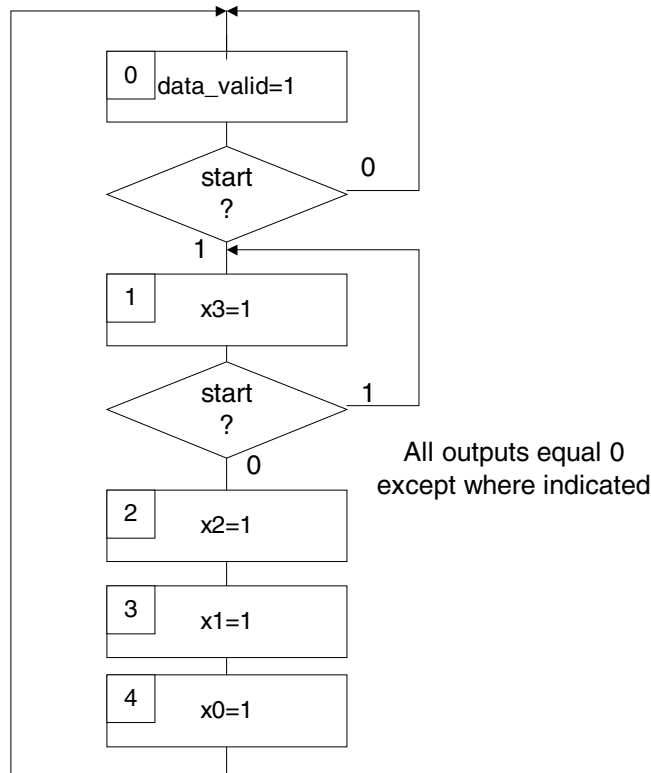
[4 marks]

Answer to Question 3

Can design one FSM or combine FSM with registers, gates etc. Here is one possible solution:



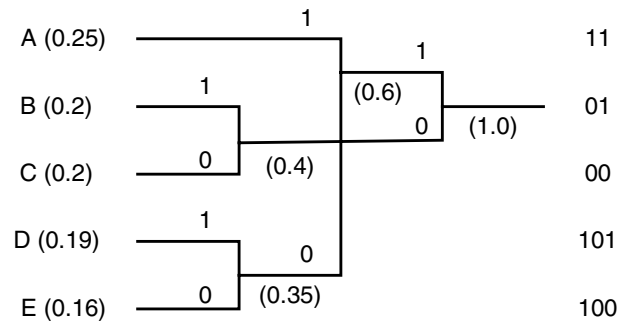
The ASM chart for the FSM is:



Alternate solution is to design a FSM which is essentially a binary decision tree with 31 states. The first state is entered when *start* goes high. The lowest 16 states determine the final *R* values. The machine should remain in the bottom state until another *start* is detected.

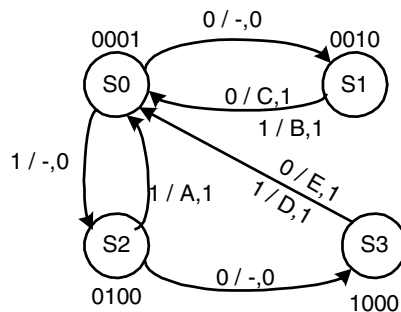
Answer to Question 4

(a)



[4 marks]

(b)



[6 marks]

(c)

| DIN | Current state (T3:0) | Next state (N3:0) | VALID | DOUT2:0 |
|-----|----------------------|-------------------|-------|---------|
| 0 | 0001 | 0010 | 0 | xxx |
| 1 | 0001 | 0100 | 0 | xxx |
| 0 | 0010 | 0001 | 1 | 010 |
| 1 | 0010 | 0001 | 1 | 001 |
| 0 | 0100 | 1000 | 0 | xxx |
| 1 | 0100 | 0001 | 1 | 000 |
| 0 | 1000 | 0001 | 1 | 100 |
| 1 | 1000 | 0001 | 1 | 011 |

$$N3 = T2 * /DIN$$

$$N2 = T0 * DIN$$

$$N1 = T0 * /DIN$$

$$DOUT2 = T3 * /DIN$$

$$N0 = VALID = T1 + T3 + T2 * DIN$$

$$DOUT0 = T1 * DIN + T3 * DIN$$

$$DOUT1 = T1 * /DIN + T3 * DIN$$

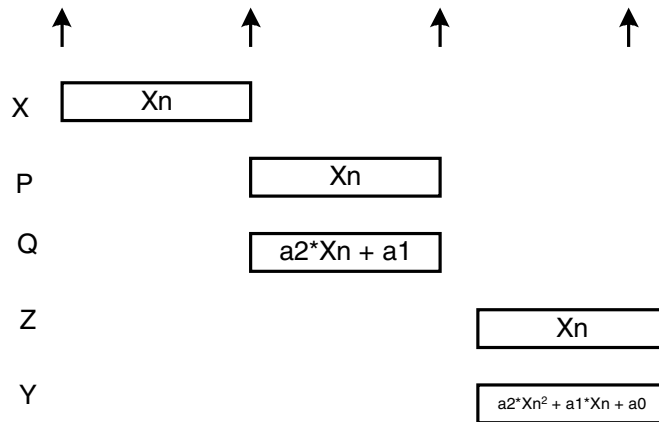
[8 marks]

(d) No equations has more than 4 input variables. Therefore only requires one logic cells to implement each equation include the state flip-flop. Therefore # of logic cell = 7.

[2 marks]

Answer to Question 5

(a)



The relationship is:

$$Y = a_2 * x_n^2 + a_1 * x_n + a_0 \text{ (two cycles delay)}$$

$$Z = x_n \text{ (two cycles delay)}$$

[12 marks]

(b)

$$Y = a_N * x_n^N + a_{N-1} * x_n^{N-1} + \dots + a_2 * x_n^2 + a_1 * x_n + a_0 \text{ (N cycles delay)}$$

[5 marks]

(c)

Clock → Pin → M2 → M3 → M5setup

$$= 10 + 25 + 10 + 5 \text{ ns} = 50 \text{ ns.}$$

Therefore Max frequency = 20 MHz for **any** value of N.

[3 marks]

Answer to Question 6

(a)

R check bits can check $2^R - R - 1$ bit errors, therefore $R=4$ for 8-bit data.
The 4 parity bits are used according to this table:

| | d7 | d6 | d5 | d4 | d3 | d2 | d1 | d0 |
|------|----|----|----|----|----|----|----|----|
| P3 | x | x | x | x | | | | |
| P2 | x | | | | x | x | x | |
| P1 | | x | x | | x | x | | x |
| P0 | | x | | x | x | | x | x |
| Code | 12 | 11 | 10 | 9 | 7 | 6 | 5 | 3 |

$$P3 = d7 \oplus d6 \oplus d5 \oplus d4$$

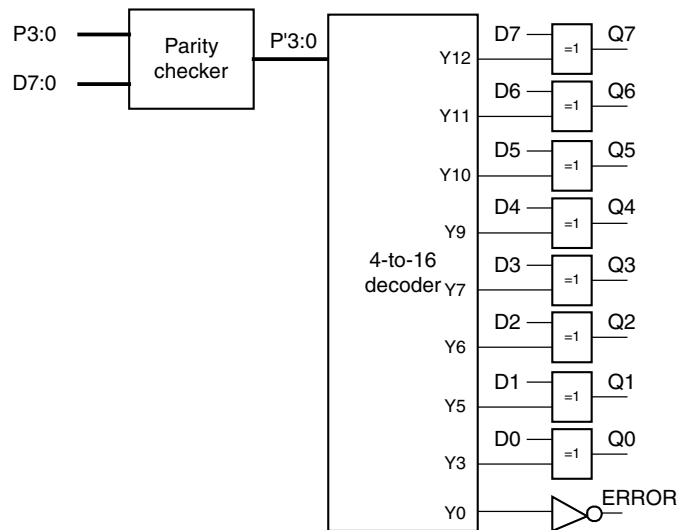
$$P2 = d7 \oplus d3 \oplus d2 \oplus d1$$

$$P1 = d6 \oplus d5 \oplus d3 \oplus d2 \oplus d0$$

$$P0 = d6 \oplus d4 \oplus d3 \oplus d1 \oplus d0$$

[6 marks]

(b)



$$P3' = d7 \oplus d6 \oplus d5 \oplus d4 \oplus P3$$

$$P2' = d7 \oplus d3 \oplus d2 \oplus d1 \oplus P2$$

$$P1' = d6 \oplus d5 \oplus d3 \oplus d2 \oplus d0 \oplus P1$$

$$P0' = d6 \oplus d4 \oplus d3 \oplus d1 \oplus d0 \oplus P0$$

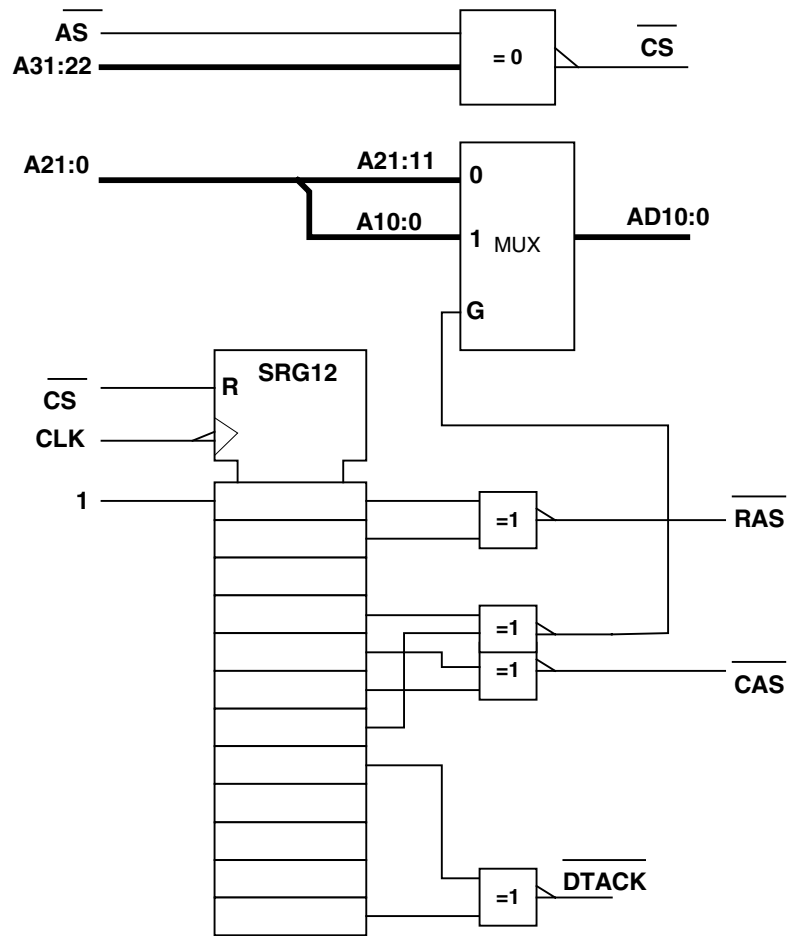
4-to-16 decoder asserts high on Y_n where $n =$ binary code of the input $P'3:0$.

For $D7:0 = 01101001$, $P3:0 = 0101$.

If bit 0 is wrong, retrieved data is 01101000. $P'3:0 = 0011$. 4-to-16 decoder output will have $Y3=1$ and all other output = 0. Therefore $ERROR = 1$ and $D0$ is inverted to give $Q7:0 = 01101001$.

[14 marks]

Answer to Question 7



[20 marks]