E3.05
ISE3.19

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2007

EEE/ISE PART III/IV: MEng, BEng and ACGI

DIGITAL SYSTEM DESIGN

Friday, 4 May 10:00 am

Time allowed: 3:00 hours

Corrected Copy
Q 6

There are SIX questions on this paper.

Answer FOUR questions.

All questions carry equal marks

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible First Marker(s) : P.Y.K. Cheung
Second Marker(s) : T.J.W. Clarke

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 X[7:0] 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. a) Figure 1.1 shows a Pseudo Random Binary Sequence (PRBS) generator implementing the primitive polynomial $1+x+x^4$.

   (i) Assuming that the value of `Q[3:0]` = 0001, list the entire PRBS sequence before it repeats itself.

   (ii) Explain why this circuit may not function properly and modify the circuit to correct the potential problem.

   [6 Marks]

   b) In a data processing system, all data are stored in memory as 4-bit digits. To ensure any single bit error can be corrected, each digit is stored with its Hamming error correction codes.

   (i) How many check bits are required for each digit? Show the Boolean relation between the data bits and each check bit.

   [4 Marks]

   (ii) If a stored value of 1010 is retrieved as 1000, demonstrate how this error can be corrected.

   [5 Marks]

   (iii) Design a digital circuit that corrects any single bit errors when a digit is read from memory. Your solution can be in the form of Boolean equations.

   [5 Marks]

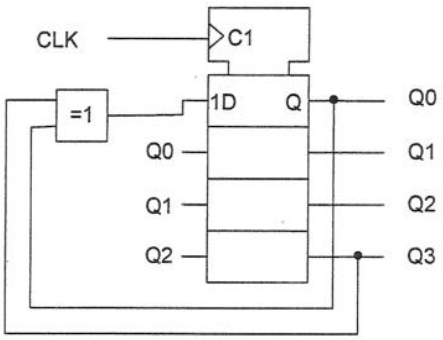


Figure 1.1

2. *Figure 2.1* shows the state transition table for a finite state machine (FSM) with two inputs X, one output Z, and eight states.

a) Using the implication chart method, reduce this to a FSM with only three states. Templates of the implication chart are provided at the end of the question section.

[12 Marks]

b) Draw the reduced state transition table.

[4 Marks]

c) A Logic Element (LE) in Altera's Cyclone-II PLD device consists of a 4-input lookup table and a register. Implement the reduced FSM using on the Cyclone-II device using one-hot state encoding. How many LEs are required to implement the FSM?

[4 Marks]

Current State	Next state		Output Z	
	Input X		Input X	
	0	1	0	1
1	8	3	0	0
2	8	6	0	0
3	8	1	1	0
4	1	8	1	0
5	4	7	0	0
6	4	2	1	0
7	4	5	1	0
8	5	4	1	0

Figure 2.1

3. a) Outline the operating principle of JTAG boundary scan. Discuss why the JTAG boundary scan test standard is important in testing modern electronic systems. Brief explain how a circuit containing JTAG compliant components can be tested for interconnection faults

[8 marks]

b) *Figure 3.1* shows part of the printed circuit board layout with two JTAG compliant components in 16-pin dual-in-line packages. Pins are designated from 1 to 16 anticlockwise. Boundary scan cells are serially connected from pin 6 (head of the scan path) down to pin 1, and pin 16 down to pin 11 (tail of the scan path). `TDI`, `TDO`, `TMS` and `TCK` are the Test-Data-In, Test-Data-Out, Test-Mode-Select and Test-Clock signals respectively used for JTAG testing. Interconnection faults on the five nets `A`, `B`, `C`, `D` and `E` could either be stuck-at faults, or bridging faults to the neighbouring tracks.

Assuming that all the pins on U1 and U2 are configured as outputs and inputs respectively, design the input test sequences to be sent to `TDI` in order to detect both stuck-at faults and bridging faults on nets `A`, `B` and `C`. Give the corresponding fault-free output sequences from `TDO`.

[12 marks]

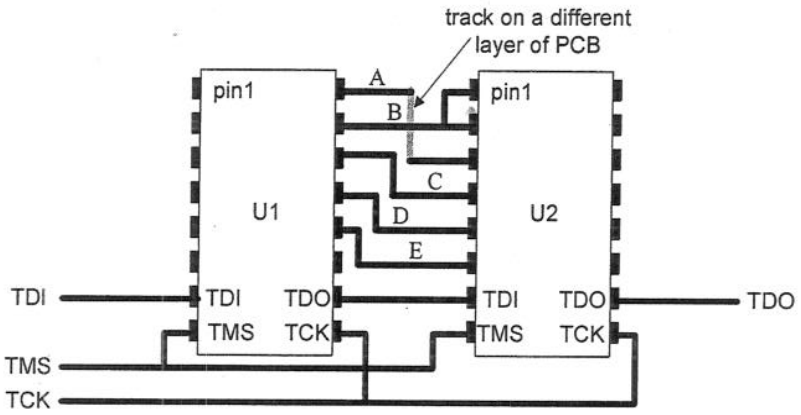


Figure 3.1

4. The operation of a 5-tap convolver is given by the equation:

$$y = 0.75x_0 + 0.25x_1 + x_2 - 0.375x_3 - 0.875x_4 \qquad (4.1)$$

where x_n for $n = 0$ to 4 are 16-bit integer data in 2's complement form.

a) Show that *equation 4.1* can be rewritten as:-

$$y = 0.125 \times \left\{ \left[\left[(x_2 - x_4) \times 2 + (x_0 - x_3) \right] \times 2 + (x_0 + x_1) \right] \times 2 + (x_3 + x_4) \right\}$$

[6 marks]

b) Hence, or otherwise, design a circuit to implement this convolver using a tree of two-input multi-bit adders and subtractors. Explain clearly how the adders and subtractors are interconnected in order to ensure that no overflow can occur. Show clearly the width of all the datapaths in your circuit and the position of the binary point at the output.

[14 marks]

5. *Figure 5.1* shows part of the circuitry needed to divide an 8 bit unsigned number by a 4 bit unsigned number `X[3:0]`. The 8 bit number is initially loaded into the register `Z[7:0]` with `CLK` held low. The division is then accomplished by applying four pulses to `CLK`.

While `CLK` is low, the adder is used to determine whether the number `Z[7:3]` is greater than `X[3:0]`. If this is the case, `SUB` goes high and `X` is subtracted from the register contents on the trailing edge of the next pulse of `CLK`.

a) Draw a table showing the binary values that are taken by `CLK`, `Y[3:0]`, `Z[7:0]`, `S[4:0]`, `SUB` and `BIT` after each transition of `CLK` during the process of dividing 104 (01101000_B) by 9 (1001_B).

[10 marks]

b) At the end of this division, what do `Z[3:0]` and `Z[7:4]` contain?

[6 marks]

c) For what initial values of `Z` and `X` will the circuit fail to give the correct answer?

[4 marks]

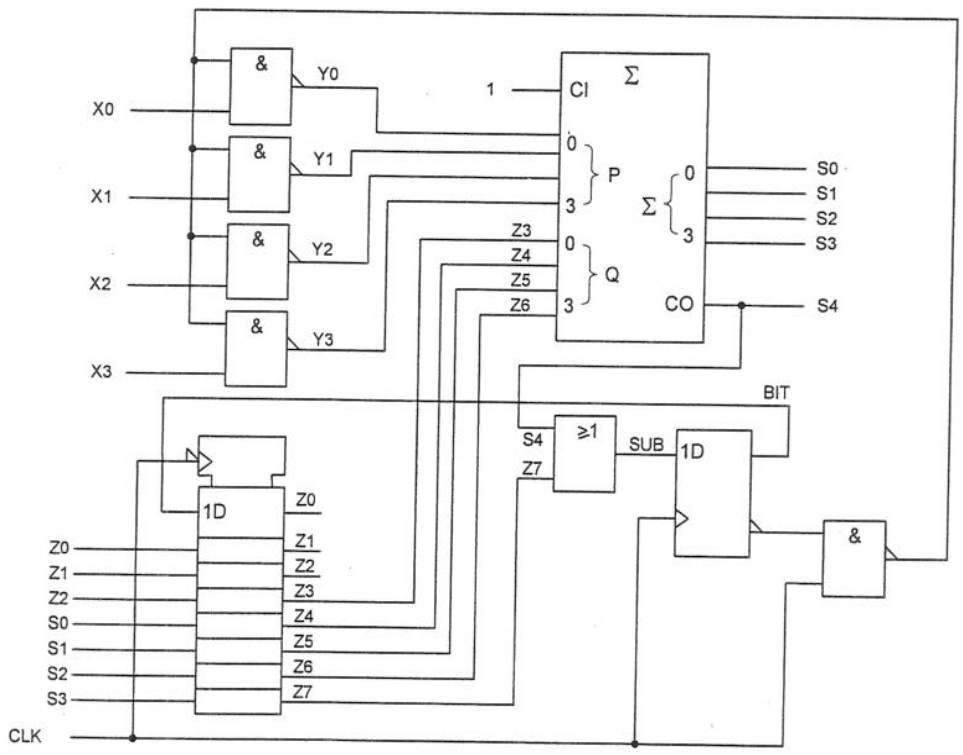


Figure 5.1

Announced 10:20

6. *Figure 6.1* depicts a 256 x 32 bit synchronous stack or last-in-first-out (LIFO) memory module that is implemented using an FPGA. The signal `reset` initializes the LIFO to an empty state. All control, address and data signals change shortly after the rising edge of `clk`.

   The LIFO has three possible operations: push, pop and no operation (nop). On the rising edge of `clk`, input data `d_in` is pushed onto the LIFO if `push` is high. Similarly output data is popped onto `d_out` if `pop` is high. The first data words read out from the LIFO is the last data stored. If the number of data words remaining in the LIFO is 511, the signal `full` goes high and any further push operations will be ignored. Similar if the number of data words stored is 0, the signal `empty` goes high and any further pop operations will be ignored. 255

   a) *Figure 6.2* depicts a programmable block RAM found in the FPGA. Each block RAM can be configured as 2048×1, 1024×2, 512×4 or 256×8 RAM. The data output can be registered or unregistered depending on how the block RAM is configured as shown in *Figure 6.2*. With the aid of a diagram, explain how you would use multiple block RAMs to implement the memory used in the LIFO.

   [4 marks]

   b) *Figure 6.3* shows a simplified block diagram of the LIFO circuit. Assuming that the counter was originally reset to 0, draw a timing diagram showing the operation of the LIFO for the sequence of operations: push, push, push, nop, pop, push, pop, push, push, push, pop, pop. Your diagram should include the signals: `clk`, `push`, `pop`, `ctr_out`, and the address read/written to the block RAM.

   [6 marks]

   c) Modify the circuit shown in *Figure 6.3* to include the LIFO full and empty detection.

   [6 marks]

   d) Estimate with justification the approximate number of logic elements (LEs) required to implement the LIFO. Each LE consists of a 4-inputs-1-output lookup table (LUT) and an optional 1-bit register.

   [4 marks]

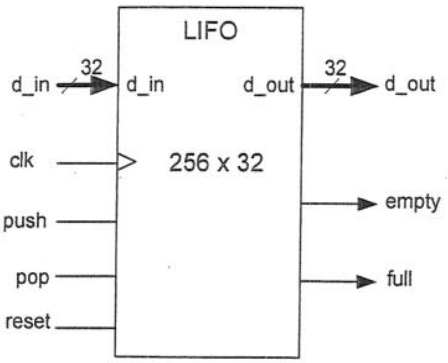


Figure 6.1

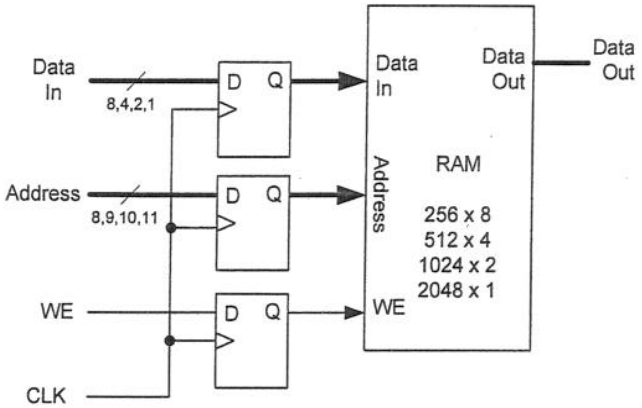


Figure 6.2

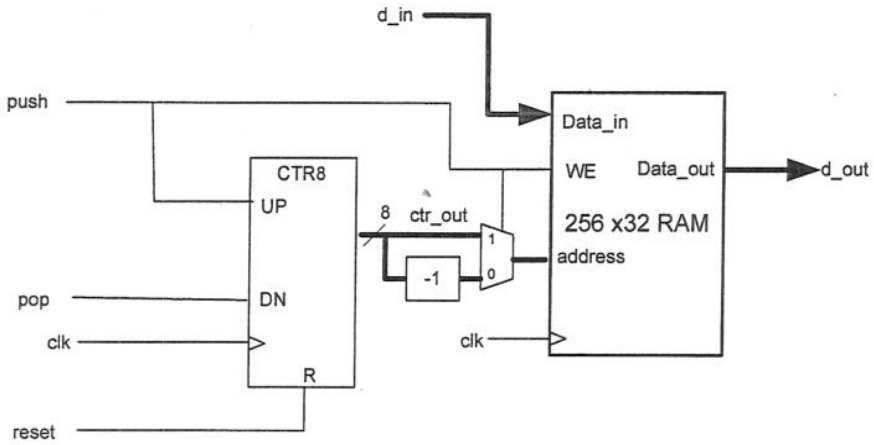
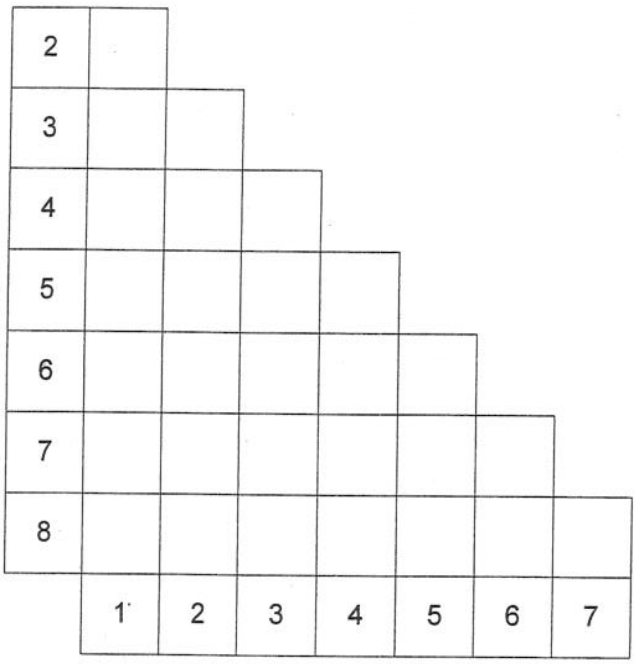


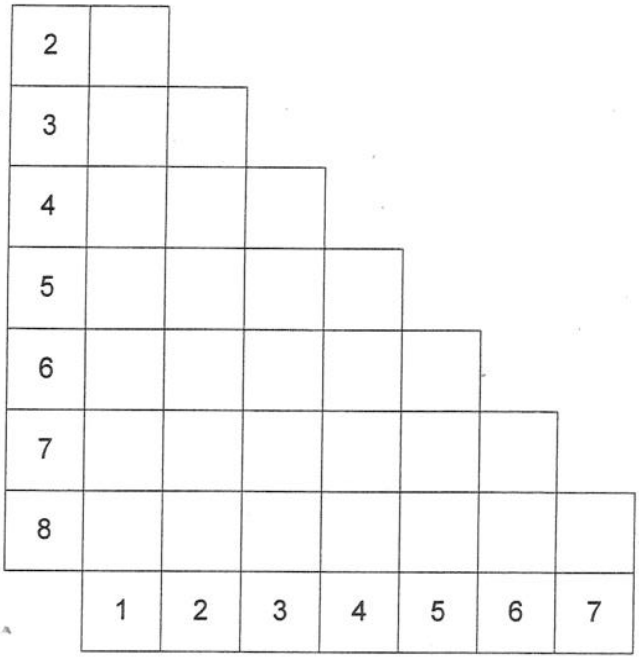
Figure 6.3

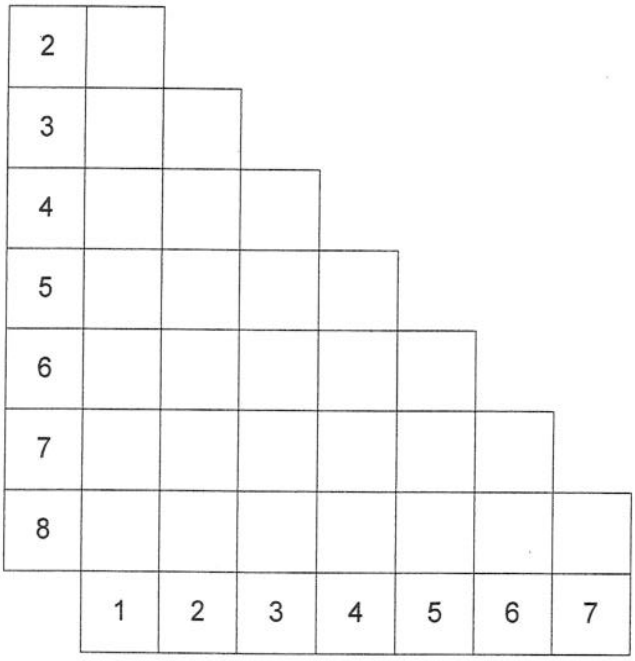
[END]

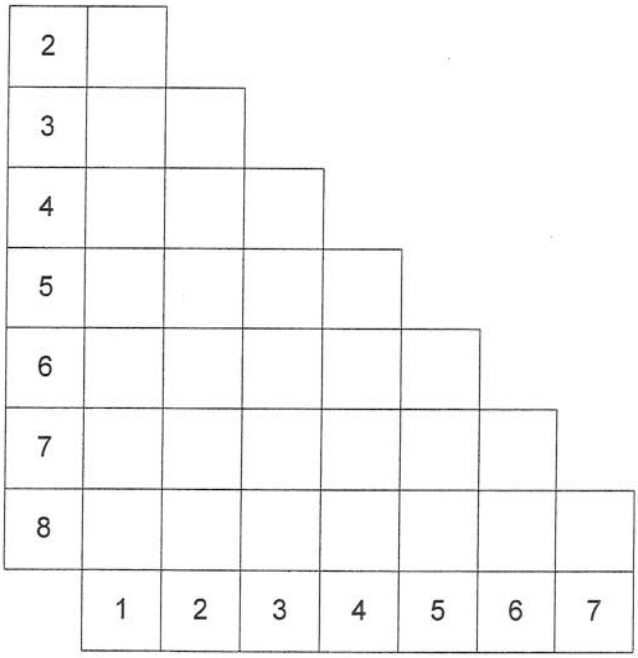
(This page is deliberately left blank.)

Implication Chart Templates for Question 2









(If you use this template, please make sure that you attach it with your answer book.)

Candidate Number: _____

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 2007

PART III and PART IV

DIGITAL SYSTEM DESIGN

SOLUTIONS

First Marker: *Peter Y. K. Cheung*

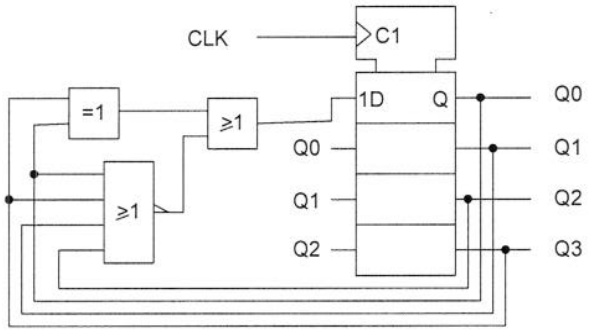
Second Marker: *Thomas Clarke*

Answer to Question 1

This question tests students' understanding of PRBS and error correcting codes (for memory).

(a) This is a maximal length sequence with 15 states. The states (in hex) are: 1, 3, 7, F, E, D, A, 5, B, 6, C, 9, 2, 4, 8, 1,…..

The circuit fails if the state is 0 (stuck in this state). Therefore when this happens, it must forces the circuit to go to state 1.



(b) (i) N check bits can check $2^N - N - 1$ data bits. Therefore we need 3 check bits.

	0	1	2	3
p0	x	x		x
p1	x		x	x
p2		x	x	x

Therefore

p0 = d0 ⊕ d1 ⊕ d3
p1 = d0 ⊕ d2 ⊕ d3
p2 = d1 ⊕ d2 ⊕ d3

[4 marks]

(ii)

If a nibble digital of 1010 is read as 1000, d1 is wrong. Parity bit derived would be p0 p1 p2 = 111 instead of 010. We can conclude that p0 and p2 are wrong. With single bit error assumption, the only possible conclusion is that d1 is wrong. Therefore d1 can be corrected.

If the nibble digital of 1010 is read as 1000, then both d1 and d3 are wrong. Parity bit derived will be 000 instead of 010.

[5 marks]

(iii)

Assuming p0', p1' and p2' are the stored parity bits, d0' to d3' are stored data bits, and p0, p1 and p2 are the derived parity bits. Then:

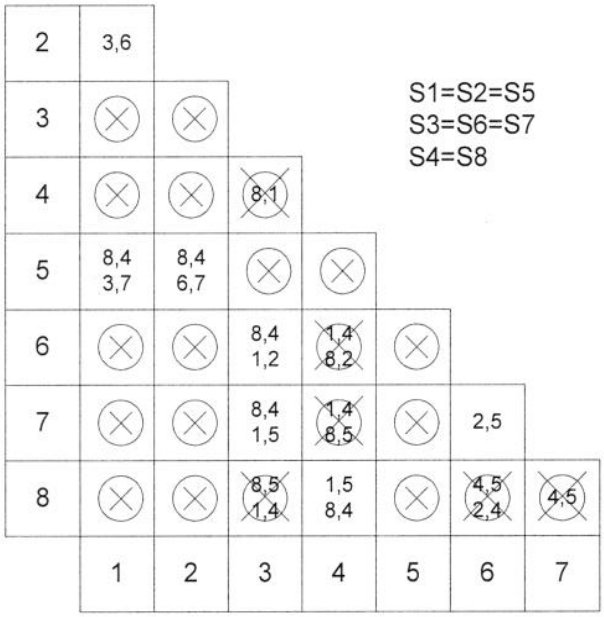
The correct data bits are:
d0 = d0' ⊕ ((p0 ⊕ p0') • (p1 ⊕ p1') • !(p2 ⊕ p2'))
d1 = d1' ⊕ ((p0 ⊕ p0') • !(p1 ⊕ p1') • (p2 ⊕ p2'))
d2 = d2' ⊕ (!(p0 ⊕ p0') • (p1 ⊕ p1') • (p2 ⊕ p2'))
d3 = d3' ⊕ ((p0 ⊕ p0') • (p1 ⊕ p1') • (p2 ⊕ p2'))

[5 marks]

Answer to Question 2

This question tests students on state reduction using implication chart method, and implementation on FPGA using one-hot encoding.

a)



[12 marks]

(b)

Current State (s2:s1:s0)	Next state (n2:n1:n0)		Output Z	
	Input X		Input X	
	0	1	0	1
1 (001)	4 (100)	3 (010)	0	0
3 (010)	4 (100)	1 (001)	1	0
4 (100)	1 (001)	4 (100)	1	0

[4 marks]

(c)

With the encoding shown above, the implementation is:

n0 = s1*X + s2*/X

n1 = s0*X

n2 = s2*x + s1*/X + x0*/X

Z = s1*/X + s2*/X

This requires 4 LEs.

[4 marks]

Answer to Question 3

a) Mostly bookwork.

Modern electronic systems have high density PCB with surface mount components. Impossible to get at any pins of devices. Therefore testing becomes very difficult if not impossible. JTAG boundary scan provides a means of accessing both the internal circuits on components as well as external circuit outside the components.

Students should explain how JTAG actually work. .....

JTAG test configuration includes EXTEST for testing external devices. Test patterns can be strobed serially into the JTAG compliant device to drive external interconnect to 1 or 0. Another JTAG compliant device would act as 'receiver' and parallel load the pattern. This is then strobed out and compare to expected pattern.

[8 marks]

b)

To detect stuck-at-0 and stuck-at-1 faults, we must drive all nets to 1 and 0 respectively. To detect shorting faults of neighbouring nets, we must drive the neighbours with different logic values. One possible solution:

(U1 is left device, U2 is right device)

U1: shift in x10101xxxxxx

U2: capture

U2: shift out and test for xxxxxx001101

U1: shift in x01010xxxxxx

U2: capture

U2: shift out and test for xxxxxx110010

Above tests will detect both stuck-at and bridging faults for all neighbouring nets except nets A and C. Therefore additional test:

U1: x00011xxxxxx

U2: capture

U2: shift out and test for xxxxxx111000

[12 marks]

Answer to Question 4

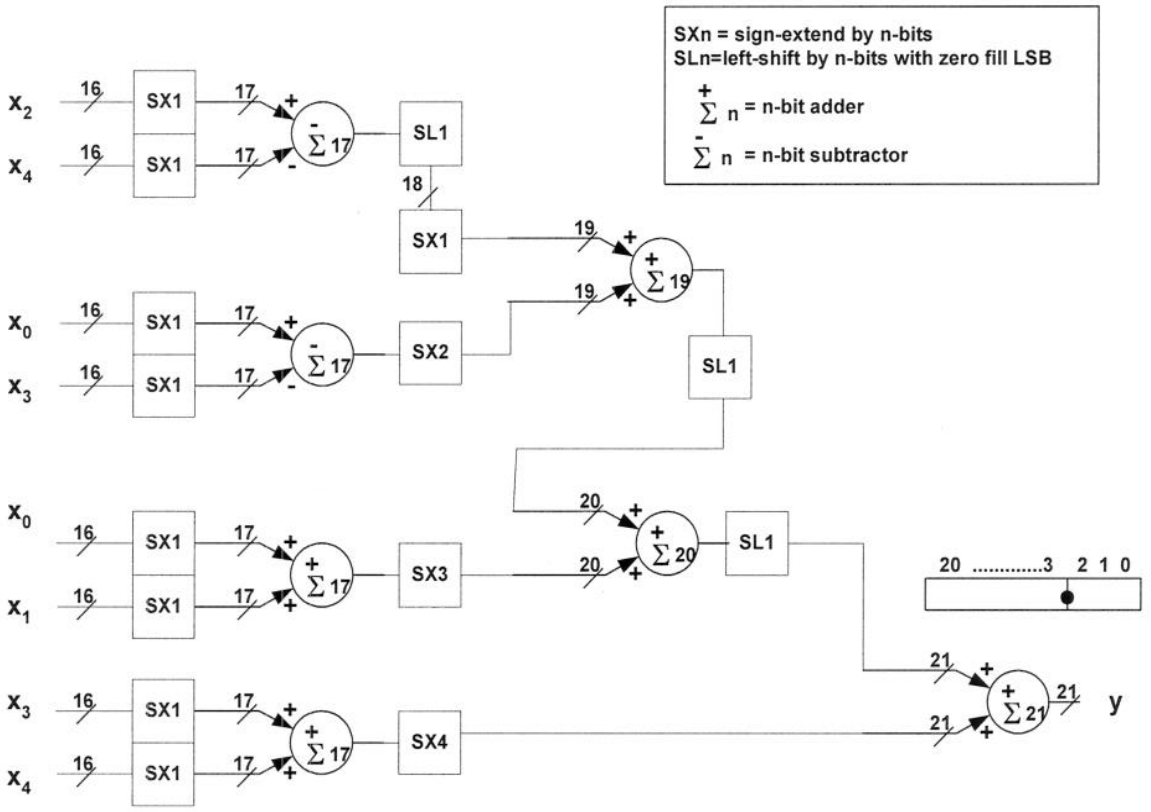
This tests students on distributed arithmetic, scaling, 2's complement computation, fractional number representation and overall architecture for sum-of-product computation.

(a) Scale all coefficients with a factor 0.125 to make them integers.

$$
\begin{aligned}
y &= 0.75x_0 + 0.25x_1 + x_2 - 0.375x_3 - 0.875x_4 \\
&= 0.125 \times \{6x_0 + 2x_1 + 8x_2 - 3x_3 - 7x_4\} \\
&= 0.125 \times \{4x_0 + 2x_0 + 2x_1 + 8x_2 - 4x_3 + x_3 - 8x_4 + x_4\} \\
&= 0.125 \times \{8x_2 - 8x_4 + 4x_0 - 4x_3 + 2x_0 + 2x_1 + x_3 + x_4\} \\
&= 0.125 \times \{[[(x_2 - x_4) \times 2 + (x_0 - x_3)] \times 2 + (x_0 + x_1)] \times 2 + (x_3 + x_4)\}
\end{aligned}
$$

[6 marks]

(b)



[14 marks]

Answer to Question 5

This question tests students' ability to analyse a fairly complicated circuit.

a)

CLK	Y[3:0]	Z[7:0]	S[4:0]	SUB	BIT
0	0110	01101000	10100	1	x
1	----	------	----	1	1
0	0110	01000001	01111	0	-
1	1111	------	11000	1	0
0	0110	10000010	00111	1	-
1	----	------	----	1	1
0	0110	01110101	10101	1	-
1	----	------	----	1	1
0	0110	01011011	10010	1	-

[10 marks]

b)

Z[3:0] contains the quotient: 1011 = 11
Z[7:4] contains the remainder: 0101 = 5

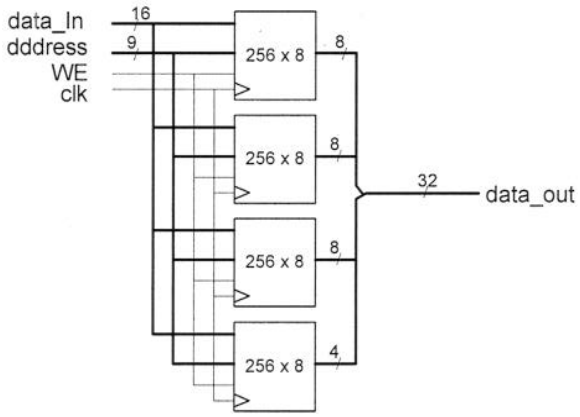
[6 marks]

c) The circuit fails to work if Z/X is greater than 15.

[4 marks]

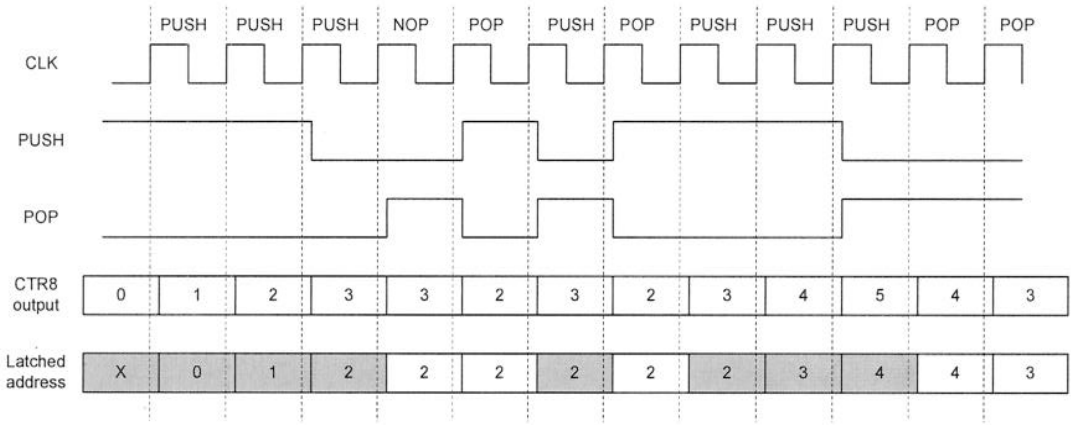
Answer to Question 6

a)



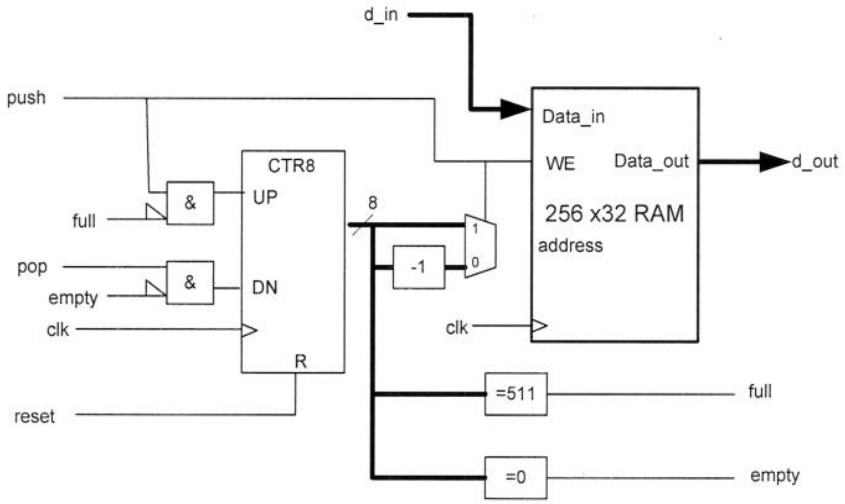
[4 marks]

b)



[6 marks]

(c)



[6 marks]

(d)

Component	Resources
Memory	4 block RAMs
Stack UP/DN counter	8 LEs
= 255 detect	2 LEs*
= 0 detect	2 LEs
-1 cct	8 LEs
MUX	8 LEs
Other gates	2 LEs
Total	**4 RAMs, 30LEs**

[4 marks]

* only 2 LEs are required due to cascade chain. Will also accept 3 LEs.

[END]