



12 Pages

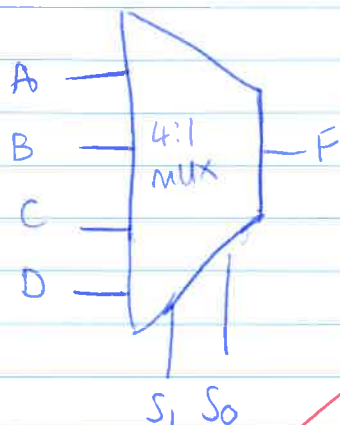
1. Please give the information requested in the boxes below.
2. Enter your name in the space provided to the right, fold over and seal down (using the label provided) to hide your name.
3. The answers must be legibly written. The Examiners can either ignore illegible answers or mark them as incorrect. If you type, the expense of typing is charged to the candidate.
4. Write your answers in ink.
5. Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
6. No part of this book is to be torn out; it must be handed in complete.
7. If more books than one are used, fasten them together using the tags available from the invigilator(s).
8. **Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.**

1	Student Number	210 8085
2	Title of Examination Paper	ELECTRONICS 2
3	Module Code	ACDA6001
4	Date and Time of Examination	10/5/23 9:00
5	Examination Venue	310
6	Seat Number	40
7	Number of Answer Books handed in	

List the numbers of the questions you have attempted	Office Use
1	20
2	18
4	20
5	17
6	17
.	
Total	92

(Gweler yr ochr arall am y fersiwn Cymraeg)

Q11a

i b. ~~A B C D~~

S_1	S_0	F
0	0	A
0	1	B
1	0	C
1	1	D

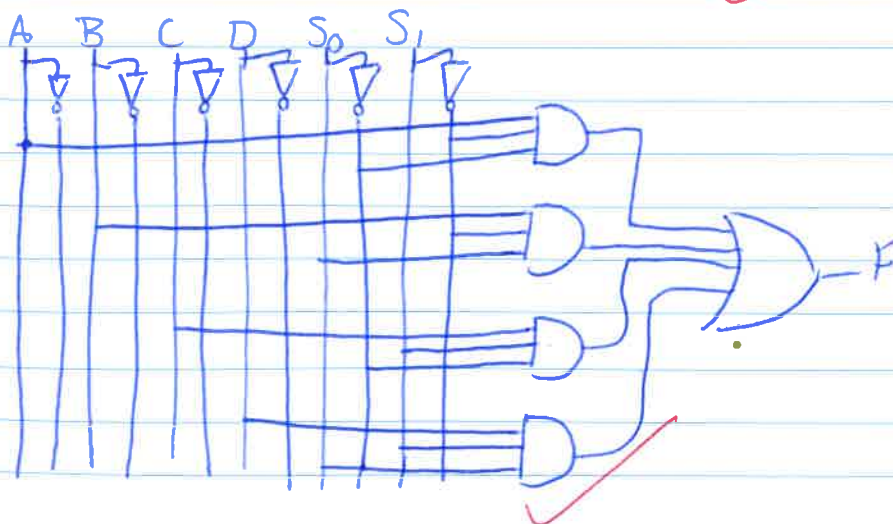
i c. ~~$A = \bar{S}_1 \bar{S}_0$~~

~~$B = \bar{S}_1 S_0$~~

~~$C = S_1 \bar{S}_0$~~

~~$D = S_1 S_0$~~

$$F = A \bar{S}_1 \bar{S}_0 + B \bar{S}_1 S_0 + C S_1 \bar{S}_0 + D S_1 S_0$$



Q1 ii library IEEE;

use IEEE.std_logic_1164.all;

entity mux is

port (A: ~~out~~ in std_logic;
 B: ~~out~~ in std_logic;
 C: ~~out~~ in std_logic;
 D: ~~out~~ in std_logic;
 S: in std_logic;
 So: in std_logic;
 F: out std_logic);

end mux;

architecture behaviour of mux is
 begin

process (A, B, C, D, S, So) is
 if (S = '0' and So = '0') then
 F <= A;

elsif (S = '0' and So = '1') then
 F <= B;

elsif (S = '1' and So = '0') then
 F <= C;

else

F <= D;

endif;

end process;

end behaviour;

1

4

5

20
 20

10

20
 20

//state assignment below

Question No.

Examiner Only

Q21i

current state	next state		output	
	x=0	x=1	x=0	x=1
a	b	a	0	0
b	e	c	0	1
c	a	c	0	0
d	b	d	0	0
e	a	c	1	1

//add all de geome a

current state	next state		output	
	x=0	x=1	x=0	x=1
a	b	a	0	0
b	e	c	0	1
c	a	c	0	0
e	a	c	1	1

//no more duplicate states

Q21ia

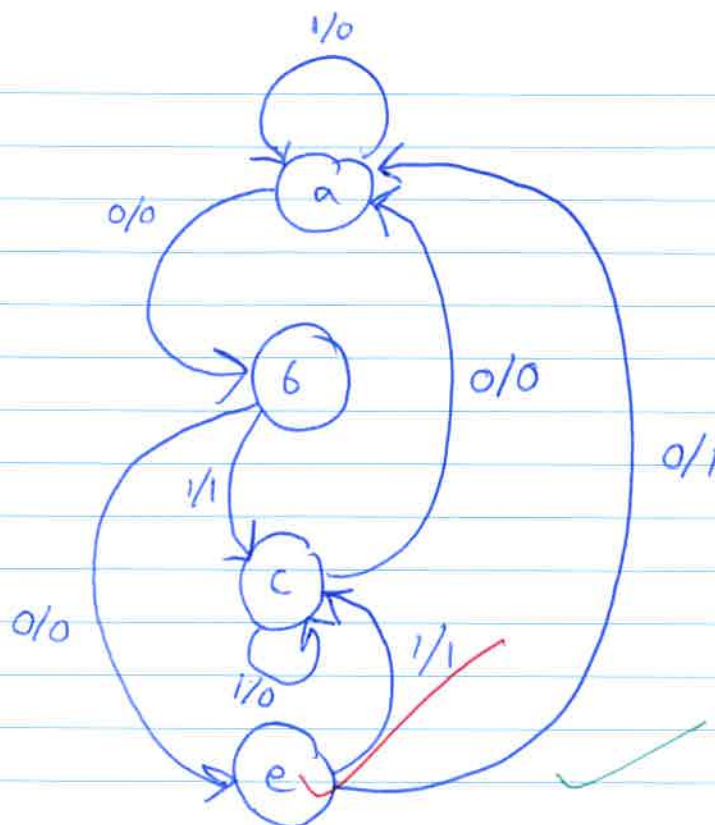
state	assignment
a	000
b	001
c	010
~~d~~	~~011~~
e	~~100~~ 011 // e is 011 if not clear
x	100
x	101

Q21ib

revised state	assignment
a	00
b	01
c	10
e	11

only 2 bits needed

Q2i



Q2ii b

state assignment on previous page

current state	input	next state	output
(a) 000	0	(b) 001	0
(a) 001	1	(a) 000	0
(b) 010	0	(e) 011	0
(b) 011	1	(c) 010	1
(c) 100	0	(a) 000	0
(c) 101	1	(c) 010	0
(e) 110	0	(a) 000	1
(e) 111	1	(c) 010	1
x	x	x	x
x	x	x	x
x	x	x	x
x	x	x	x
x	x	x	x

Only 2 bits.

current states on inputs m_1, m_0 and x

Excitation table

current state	(x) input	next state	output (y)	flip flop	d2 d1 d0
(a) 000	0	(b) 001	0	d2 d1 d0	0 0 1
(a) 000	1	(a) 000	0		0 0 0
(b) 001	0	(e) 001	0		0 1 1
(b) 001	1	(c) 010	1		0 1 0
(c) 010	0	(a) 000	0		0 0 0
(c) 010	1	(c) 010	0		0 1 0
(e) 011	0	(a) 000	1		0 0 0
(e) 011	1	(c) 010	1		0 1 0
x 100	0	x	x		x
x 100	1	x	x		x
x 101	0	x	x		x
x 101	1	x	x		x
x 110	0	x	x		x
x 110	1	x	x		x
x 111	0	x	x		x
x 111	1	x	x		x

1/3 flip flops required.

2

d. output expressions. KMAPS required

	$m_1 m_0$	00	01	11	10
$x m_0$	00	0	0	1	0
	01	x	x	x	x
	11	x	x	x	x
	10	0	1	1	0

$$y = m_1 m_0 + x m_0$$

	$m_1 m_0$	00	01	11	10
$x m_0$	00	0	0	0	0
	01	x	x	x	x
	11	x	x	x	x
	10	0	0	0	0

$$d_2 = 0$$

/actually not required

$m_1 m_0$	00	01	11	10
d_1	1	0	0	0
d_0	x	x	x	x
m_1	x	x	x	x
m_0	0	1	1	1

$$d_1 = \bar{m}_1 m_0 + x m_1$$

$m_1 m_0$	00	01	11	10
d_0	1	1	0	0
d_1	x	x	x	x
m_1	x	x	x	x
m_0	0	0	0	0

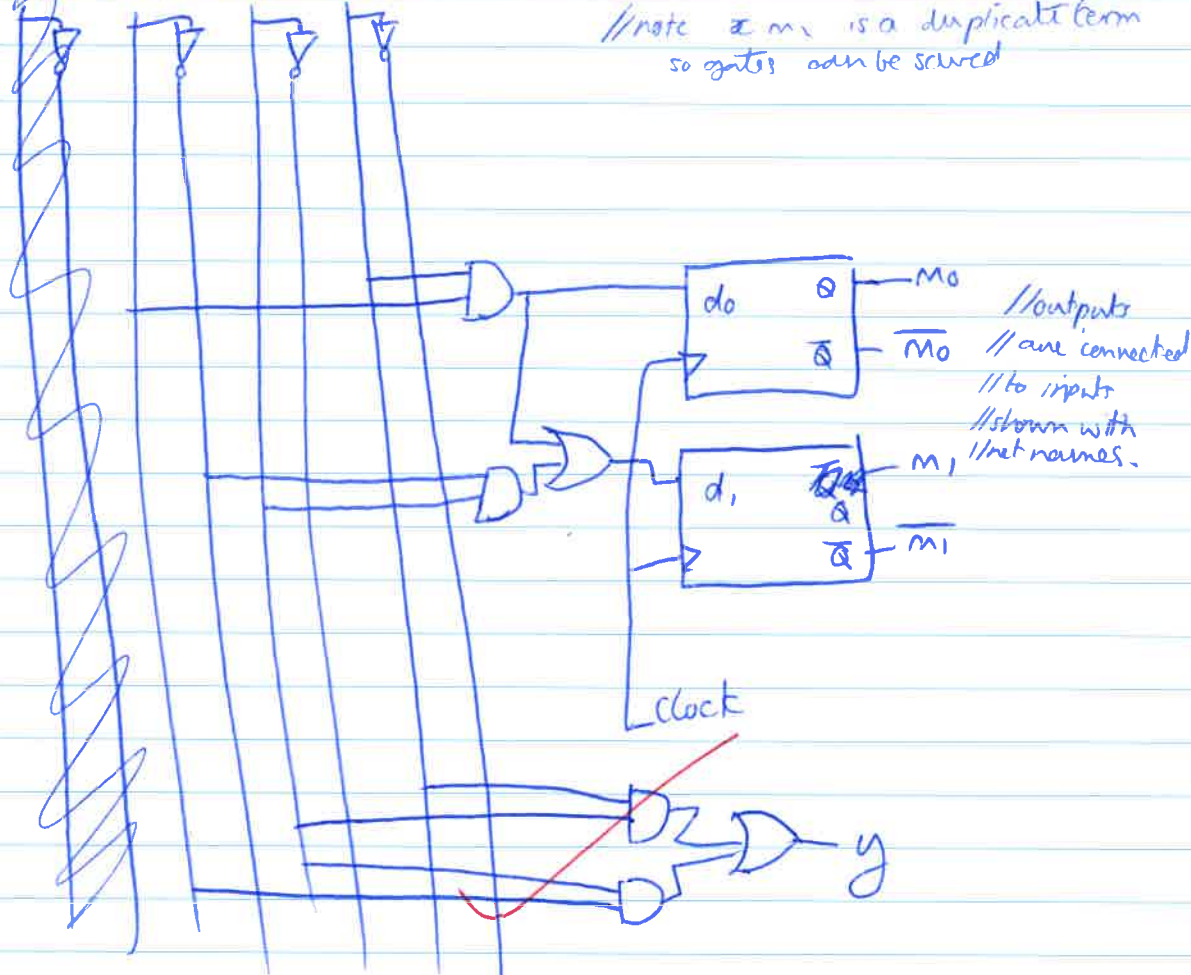
$$d_0 = x m_1$$

3

// output expressions ^{shown} in boxes

m_1, m_0, x

// note $x m_1$ is a duplicate term so gates can be saved



4

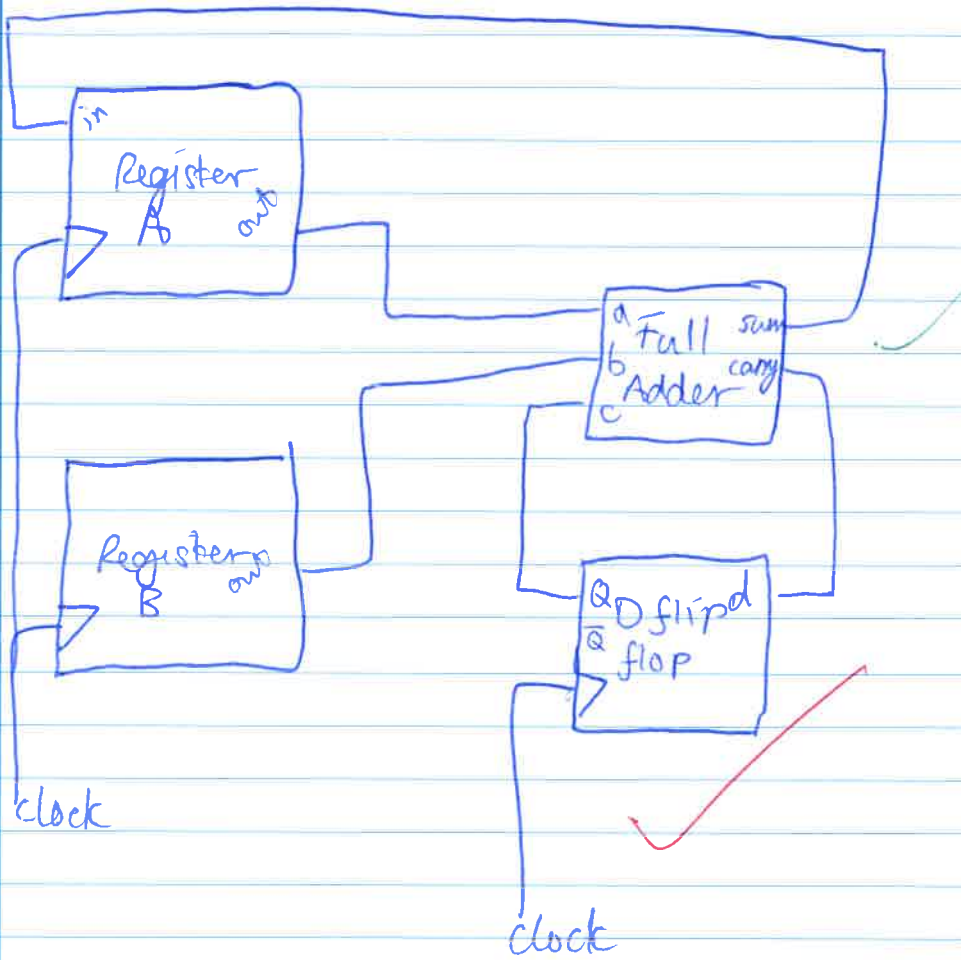
1

18/20

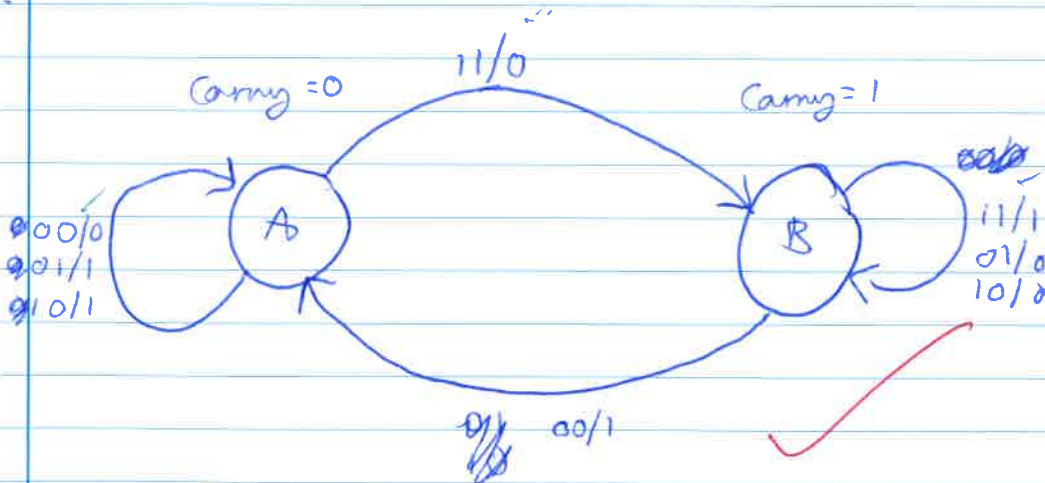
Q2 = 18

// noticed mistake when working through question, only 2 flip flops required.

Q4. Serial Adder
a. block diagram



b.



FA Truth table

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

6. ~~Present State~~ ~~Next State~~

Present State	Input	Next State	output
A	00	A	0 ✓
A	01	A	1 ✓
A	10	A	1 ✓
A	11	B	0 ✓
B	00	A	1 ✓
B	01	B	0 ✓
B	10	B	0 ✓
B	11	B	1 ✓

State assignment

A = 0

A = 0

B = 1

~~excitation table~~

Plaser

d excitation table

Present State (m_0)	Input (x_1, x_0)	Next State	Output (Z)	Flip Flops (d_0)
(A) 0	00	(A) 0	0	0
(A) 0	01	(A) 0	1	0
(A) 0	10	(A) 0	1	0
(A) 0	11	(B) 1	0	1
(B) 1	00	(A) 0	1	0
(B) 1	01	(B) 1	0	1
(B) 1	10	(B) 1	0	1
(B) 1	11	(B) 1	1	1

(Z)

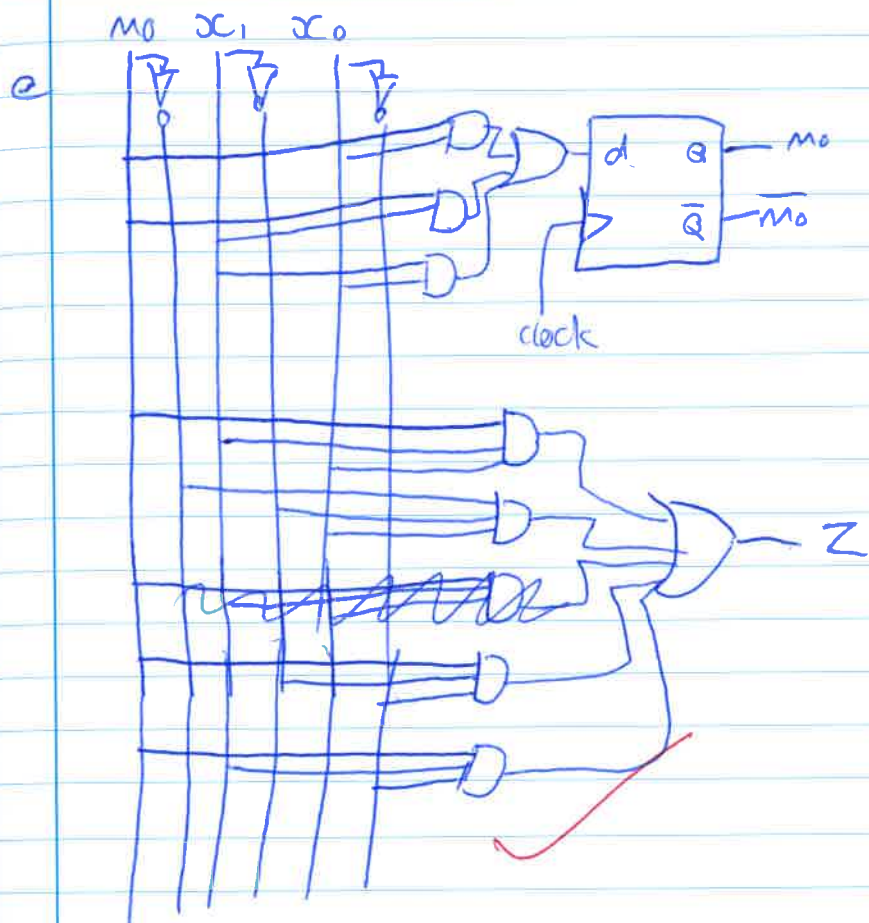
$m_0 \backslash x_1, x_0$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$Z = m_0 \bar{x}_1 \bar{x}_0 + \bar{m}_0 \bar{x}_1 x_0 + m_0 x_1 x_0 + m_0 x_1 \bar{x}_0$$

(d_0)

$m_0 \backslash x_1, x_0$	00	01	11	10
0			1	
1		1	1	1

$$d_0 = m_0 x_0 + m_0 x_1 + x_1 x_0$$



4

4

20/20

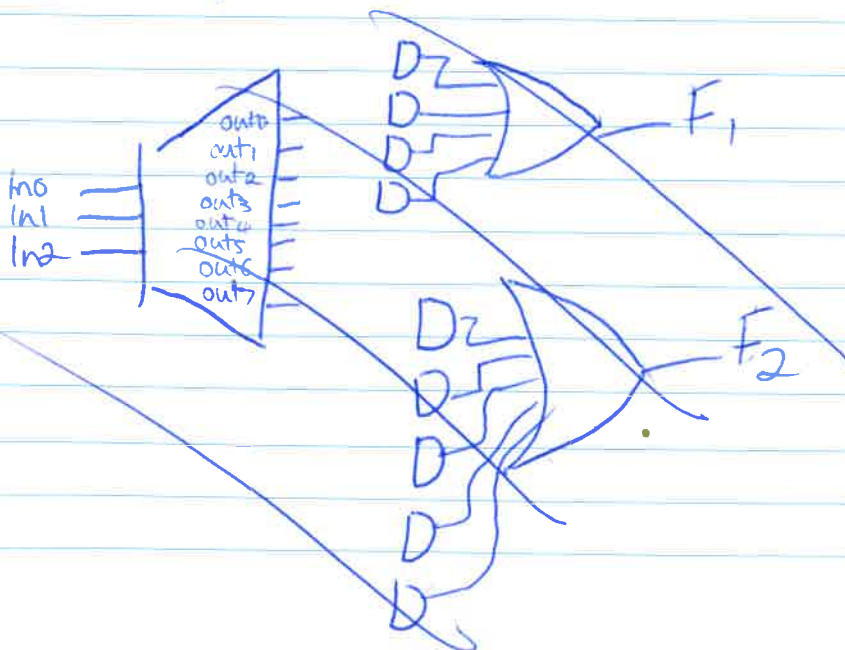
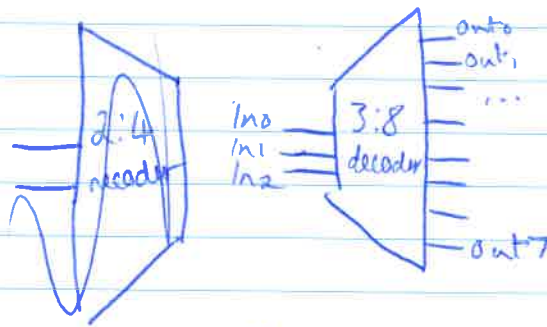
20
24

Q5a Truth table

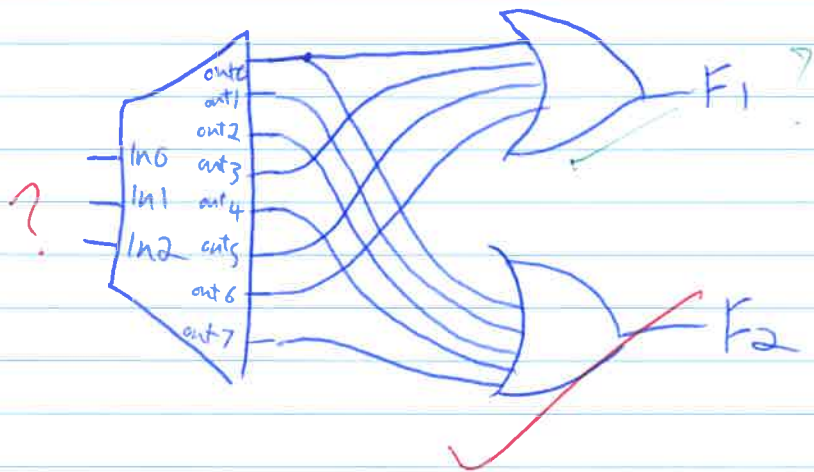
	x	y	z	F ₁	F ₂
(0)	0	0	0	1	1
(1)	0	0	1	0	1
(2)	0	1	0	0	1
(3)	0	1	1	1	0
(4)	1	0	0	0	1
(5)	1	0	1	1	0
(6)	1	1	0	1	0
(7)	1	1	1	0	1

b. $F_1 = \bar{x}\bar{y}\bar{z} + \bar{x}yz + x\bar{y}z + xy\bar{z}$

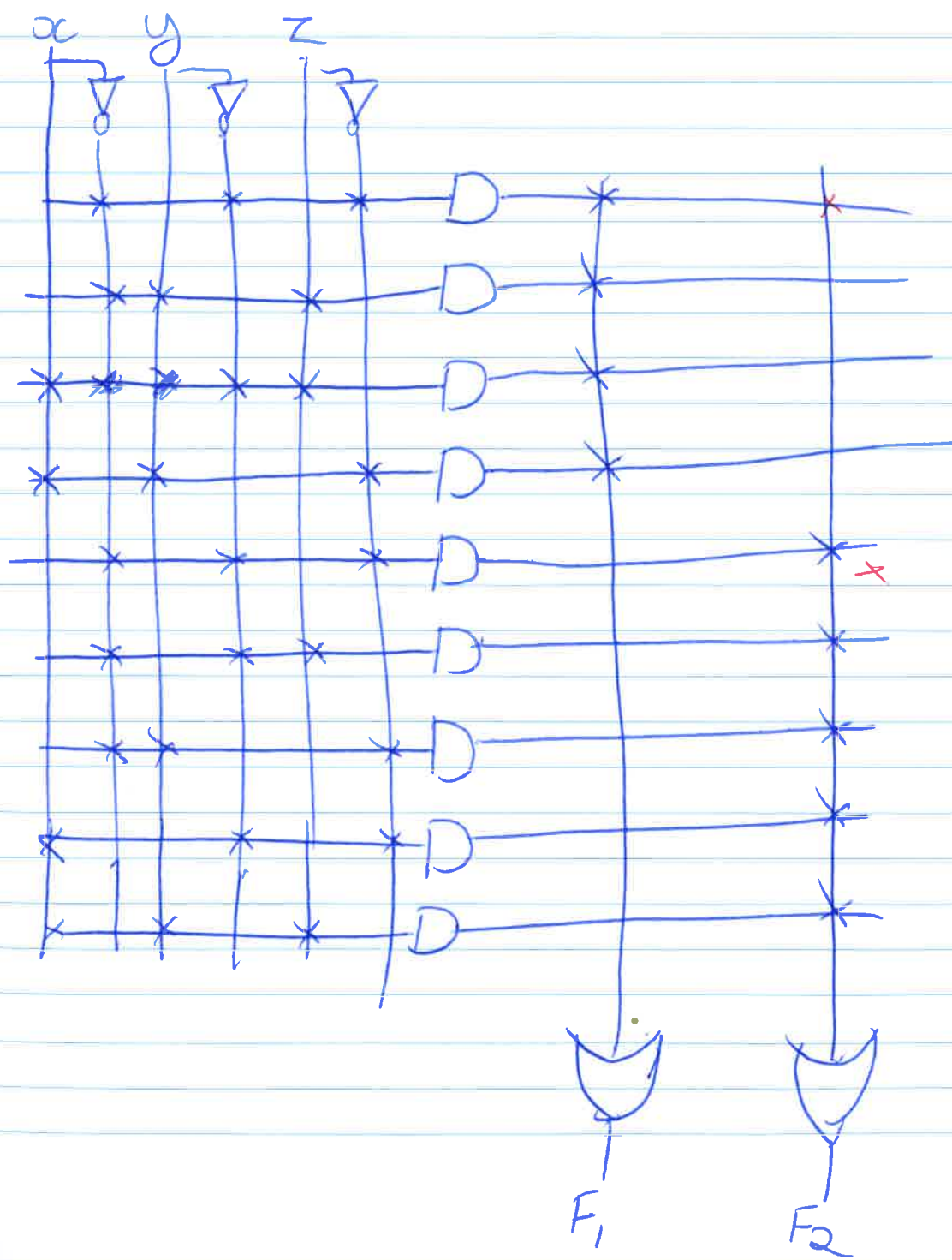
$F_2 = \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$



Q50



d. PLA design.



5
9

17
20

Q5 (17)

Q6.
Q3

$$\begin{array}{r} 5(0101) \\ -5(1011) \end{array}$$

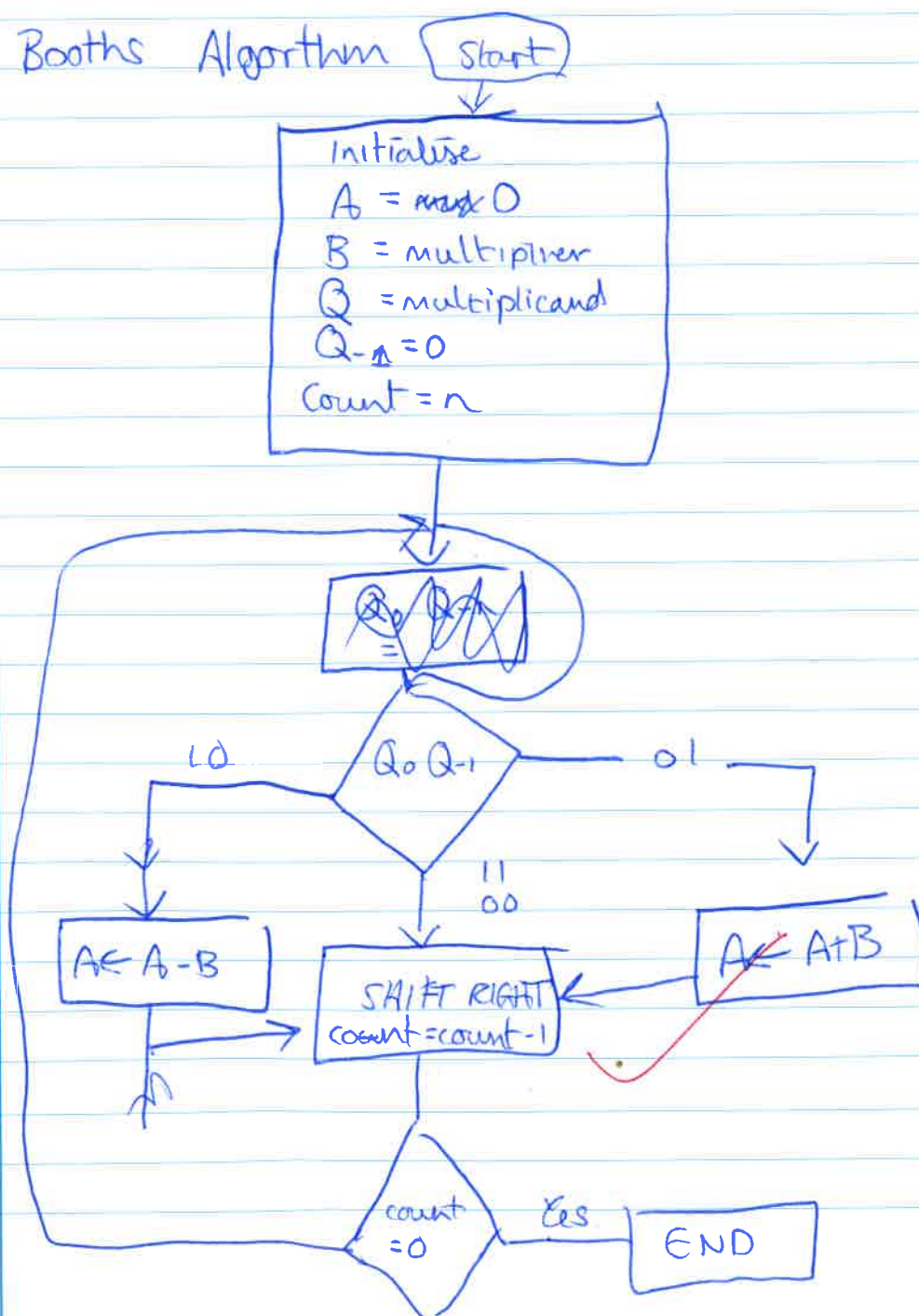
$$4(000100)$$

$$-5 \times 4 = -20$$

$$\begin{array}{r} 0100 \\ \times 1011 \\ \hline 0100 \\ 0100 \times \\ 0000 \times \times \\ 0100 \times \times \times \\ \hline 0101100 \end{array}$$

$$\begin{array}{r} 168421 \\ 20 = 10100 \\ -20 = 01011 \end{array}$$

b. Booths Algorithm



//multiplier $\rightarrow -5$
 //multiplicand $\rightarrow 4$

Question No.

Examiner Only

Count	A	Q	Q-n	Note
4	0000	0100	0	INITIALISE
3	0000	0010	0	SHIFT RIGHT
2	0000	0001	0	shift right
2	0101	0001	0	A $\leftarrow$ A-B
1	0010	1000	1	shift right
1	1101	1000	1	A $\leftarrow$ A+B
0	1110	1100	0	shift right

PRODUCT 1110 1100

// I THINK A MISTAKE HAS BEEN MADE IN ORIGINAL CALCULATION

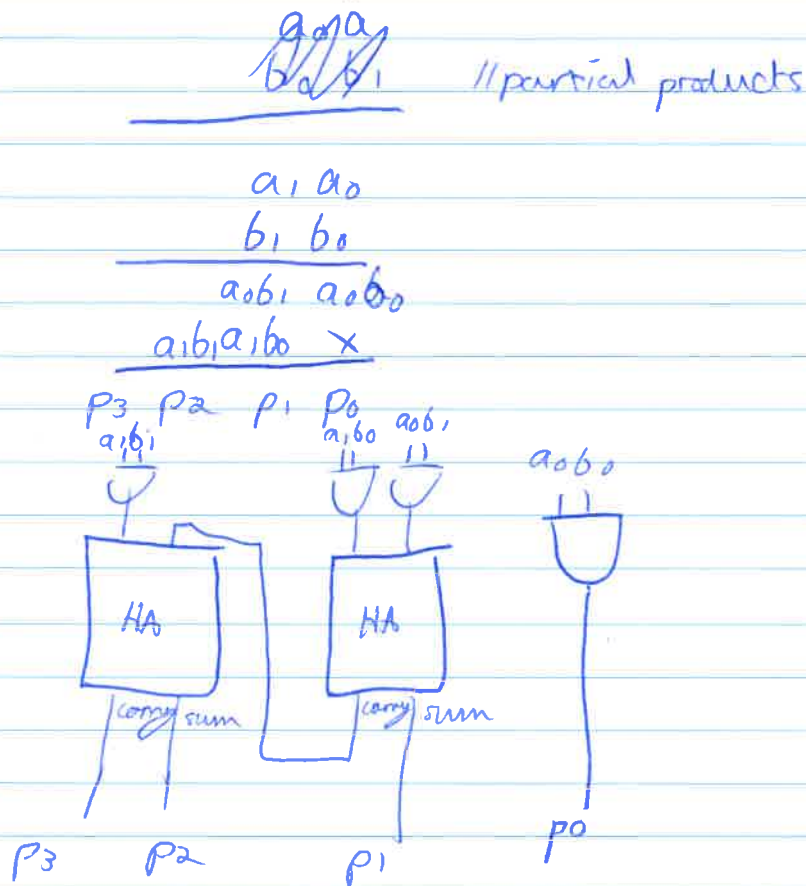
5

5

~~9~~

ii) VHDL for 2x2 multiplier

$$a \times b$$



HA

carry
sum = a and b

sum = a xor b

Not required

library IEEE

use IEEE.std_logic_1164.all

entity multiplier is

port (a: in std_logic;
b: in std_logic;
c: in std_logic;
sum: out std_logic;
carry: out std_logic);

end multiplier;

architecture dataflow of multiplier is

begin

sum

port (a: in std_logic_array (7 downto 0);
b: in std_logic_array (1 downto 0);
c: out std_logic_array (3 downto 0);

end multiplier;

architecture dataflow of multiplier is

begin

$p(0) \leq a(0) \text{ and } b(0);$

$p(1) \leq (a(0) \text{ and } b(1)) \text{ xor } (a(1) \text{ and } b(0));$

$p(2) \leq ((a(0) \text{ and } b(1) \text{ and } (a(1) \text{ and } b(0))) \text{ xor } (a(1) \text{ and } b(1)));$

$p(3) \leq ((a(0) \text{ and } b(1) \text{ and } (a(1) \text{ and } b(0)) \text{ and } (a(1) \text{ and } b(1)));$

end dataflow;

1

2

2

2

5

17
20

9

17