Paper Number(s): **E2.1**

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE
UNIVERSITY OF LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2005

EEE/ISE PART II: MEng, BEng and ACGI

DIGITAL ELECTRONICS 2

Friday, 10 June 2:00 pm

There are FOUR questions on this paper.

Q1 is compulsory.
Answer Q1 and any two of questions 2-4.
Q1 carries 40% of the marks. Questions 2 to 4 carry equal marks (30% each).

Time allowed: 2:00 hours

Examiners responsible:

First Marker(s): D.M. Brookes D.M. Brookes

Second Marker(s): T.J.W. Clarke T.J.W. Clarke

Information for Candidates:

Notation: *Unless explicitly indicated otherwise, digital circuits throughout this paper are drawn with their inputs on the left and their outputs on the right. The notation X2:0 denotes the three-bit number X2, X1 and X0. The least significant bit of a binary number is always designated bit 0.*

1. (a) In the state machine shown in *Figure 1.1*, the state is represented by the value of the unsigned 2-bit number Q1:0. The circuit is initially in state 0. Draw the state diagram for the circuit and complete the timing diagram in the figure by showing the state of the circuit during each clock cycle and the waveform of X. Do not attempt to show gate propagation delays on your timing diagram. [8]

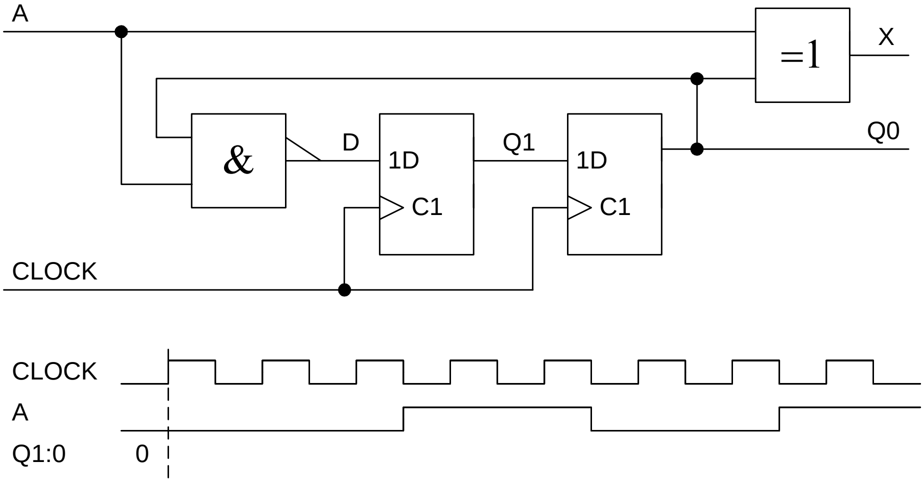


Figure 1.1

(b) In the circuit of *Figure 1.2* the propagation delay of the flip-flops is $t_p = 8\text{ ns}$ while the setup and hold times are $t_s = 5\text{ ns}$ and $t_h = 2\text{ ns}$ respectively. The inverters have a propagation delay in the range $15\text{ ns} < t_g < 25\text{ ns}$. The clock signal C is symmetrical with period T. Write down the setup and hold inequalities that apply to the rightmost flip-flop and hence find the maximum clock frequency for the circuit. [8]

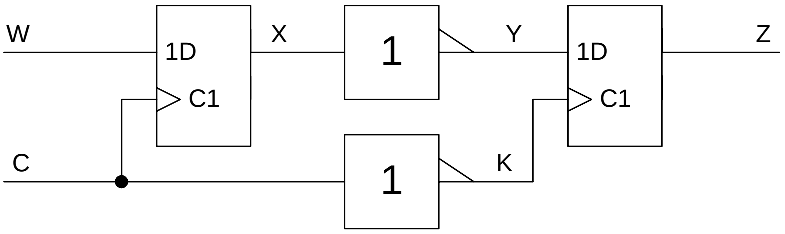


Figure 1.2

(c) *Figure 1.3* shows a digital-to-analogue converter whose input, *x*, is a 3-bit number in the range 0 to 7 and whose output voltage is *V* = *kx*. State the value of *x* that corresponds to the switch positions shown in the figure. Determine the value of the proportionality constant, *k*. [8]

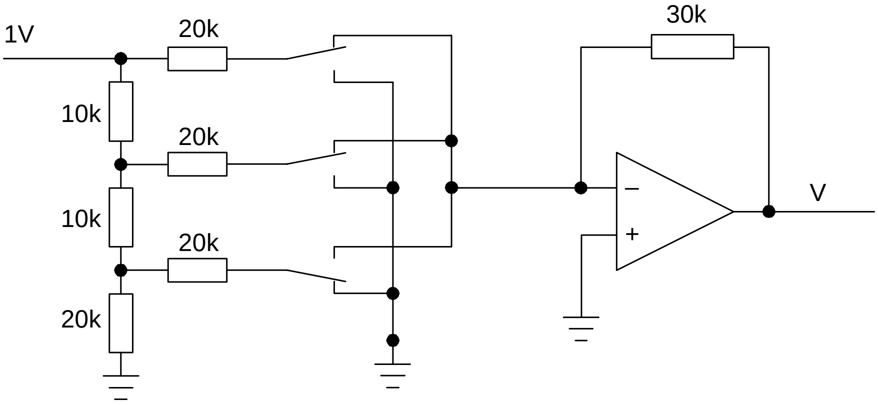


Figure 1.3

(d) *Figure 1.4* shows the circuit of a 3-bit carry-lookahead adder. The carry outputs from each full-adder are $\mathrm{CG} = \mathrm{P} \cdot \mathrm{Q}$ and $\mathrm{CGP} = \mathrm{P} + \mathrm{Q}$. Give Boolean expressions for the logic needed in the blocks labelled “C0 Logic” and “C1 Logic”. [8]

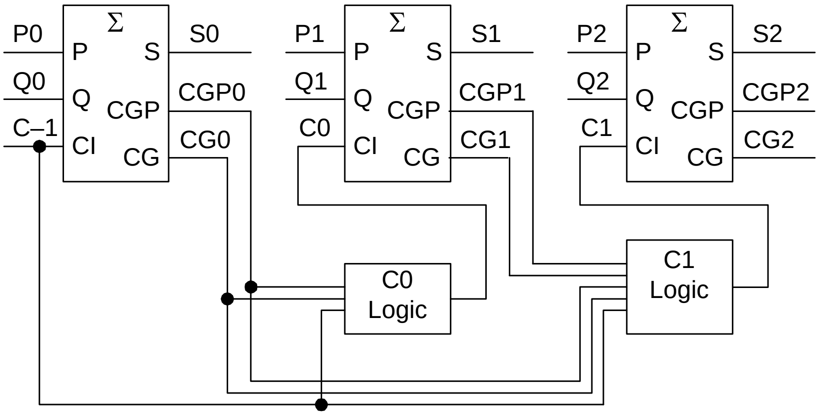


Figure 1.4

(e) *Figure 1.5* shows part of a microprocessor system containing an 8 kilobyte read-only memory (ROM). State how many address inputs will be present on the ROM. Give a Boolean expression for CE so that the ROM will be enabled only when CLOCK and READ are both high and the address A15:0 lies in the hexadecimal range $C000 to $DFFF. [8]

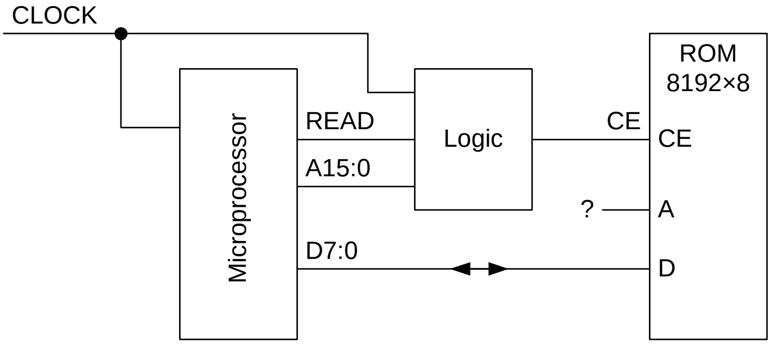


Figure 1.5

2. *Figure 2.1* shows a circuit that allows two data sources, A and B, to send information over a shared data line DOUT. DA and DB are the data signals from A and B respectively and the multiplexer determines which one is connected to DOUT. The two sources request access to the DOUT output by taking RA or RB high respectively and the controlling state machine takes GA or GB high to indicate that access has been granted. All signals change just after the CLOCK rising edge.

The contents of the logic block are given by:

$$\mathrm{NS1} = \mathrm{SELB} = \mathrm{S1}\cdot\overline{\mathrm{RA}} + \overline{\mathrm{RA}}\cdot\mathrm{RB} + \mathrm{S0}\cdot\mathrm{RB}$$

$$\mathrm{NS0} = \overline{\mathrm{S1}}\cdot\overline{\mathrm{RA}} + \overline{\mathrm{RA}}\cdot\mathrm{RB} + \mathrm{S0}\cdot\mathrm{RB}$$

$$\mathrm{GA} = \mathrm{RA}\cdot\overline{\mathrm{RB}} + \overline{\mathrm{S0}}\cdot\mathrm{RA}$$

$$\mathrm{GB} = \overline{\mathrm{RA}}\cdot\mathrm{RB} + \mathrm{S0}\cdot\mathrm{RB}$$

(a) Draw a state diagram for the circuit indicating the state transitions and the outputs in each state. The states should be labelled 0 to 3 according to the value of S1:0. [12]

(b) Complete the timing diagram shown in the figure showing the state occupied during each clock cycle and the waveforms of GA, GB and SELB. The CLOCK rising edges are marked in the diagram and the clock cycles have been numbered for your convenience. The initial state is 0 as shown. [12]

(c) Explain what happens (i) if both sources make a request simultaneously and (ii) if one source requests access to DOUT while the other is in the middle of using it. [6]

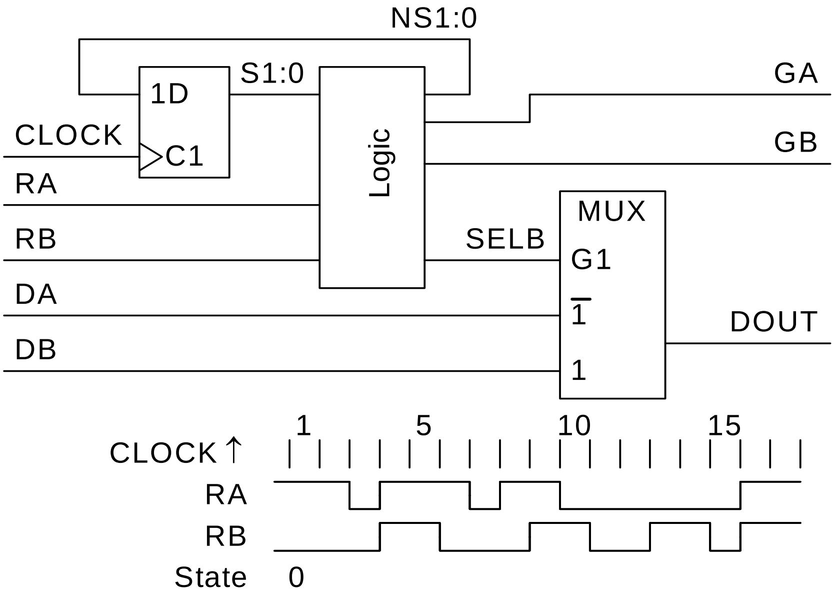


Figure 2.1

3. The circuit of *Figure 3.1* contains: a 64 byte random access memory (RAM), an 8-bit binary counter (CTR8), a 6-bit adder (Σ), a 6-bit multiplexer (MUX), a flip-flop and two 8-bit registers. The registers store a new value on the rising edge of the clock, C, but only when the enable input, EN is high just before the rising edge. Tristate outputs are marked with ∇ and enabled by the OE input. Data is written into the RAM when $\overline{\text{WE}}$ is low and read out when OE is high.

Only the six most significant output bits of the counter, Q7:2 are taken to the adder and the multiplexer; the two least significant bits are used to generate the signals F=Q1+Q0 and L=Q1·Q0. All signals transitions occur just after the rising edge of C.

(a) Assuming that N5:0 = 0 and that the counter initially contains the value Q7:0 = 18, draw a timing diagram covering eleven clock cycles and showing the values of Q7:0, A5:0 and the waveforms of Q1, Q0, F, L and $\overline{\text{WE}}$. You should indicate only the rising edges of C rather than its full waveform. [14]

(b) Indicate on your diagram the times when (i) the value X7:0 is stored, (ii) D7:0 it is written into RAM, (iii) D7:0 it is read back from RAM and (iv) Y7:0 changes to a new value. [6]

(c) If X7:0 changes only on the rising edge of Q1, determine the number of clock cycles delay between X7:0 and Y7:0 when N5:0 is equal to (i) 0, (ii) 63, (iii) 1. [10]

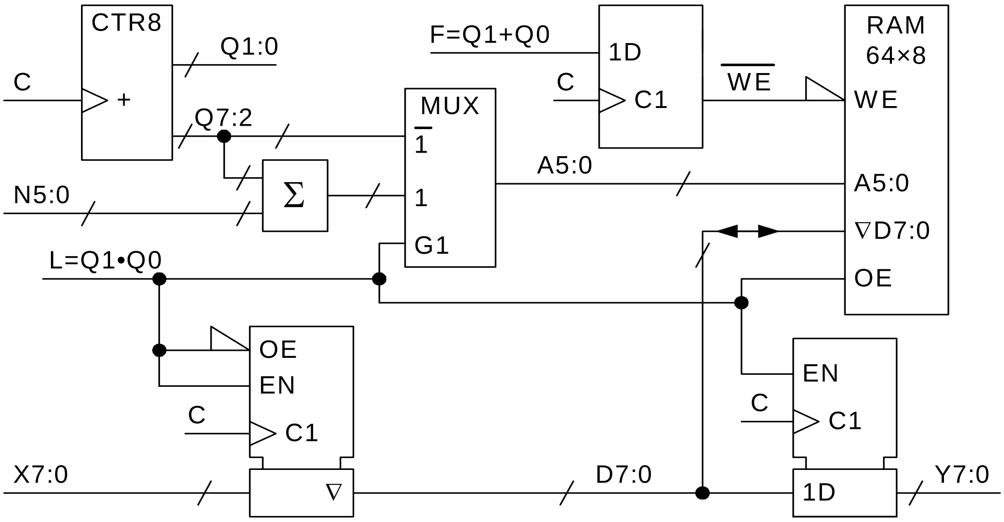


Figure 3.1

4. (a) The input to the circuit of *Figure 4.1* is an 8-bit unsigned number X7:0. The notation $[0,0,X_{7:2}]$ used for the Q input to the adder denotes an 8-bit value whose two most significant bits are 0 and whose remaining six bits are X7, X6, …, X2. Using the same notation, the 12-bit output is formed as Y11:0 = $[C_7, S_{7:0}, X_{1:0}, 0]$ where C7 is the Carry Out from the most significant adder stage. Show that the value of Y11:0 is ten times the value of X7:0. [6]

(b) In the circuit of *Figure 4.2*, X7:0 is again an unsigned 8-bit number. The notation $\overline{X7:0}$ denotes the eight bits $[\overline{X7}, \overline{X6}, \ldots, \overline{X0}]$. Showing your reasoning clearly, determine the value of Z11:0 in terms of X7:0. Determine the maximum and minimum possible values of the adder output S10:0. [7]

(c) By combining the principles illustrated in the previous parts of the question or otherwise, design a circuit using only adders to multiply an 8-bit unsigned number X7:0 by 156 (= 128+32–4 = 10011100_2). You may assume that the inverted input bits $\overline{X7:0}$ are available. You should design your circuit to minimize the total number of adder bits required. Show clearly the input and output signals for each adder using the same notation as that used in *Figure 4.1* and *Figure 4.2*. [9]

(d) The circuit of *Figure 4.3* shows an alternative circuit that multiplies by 156. The circuit combines a 14-bit full adder with a 14-bit carry-save adder whose outputs are given by $\mathrm{S}i = \mathrm{P}i \oplus \mathrm{Q}i \oplus \mathrm{R}i$ and $\mathrm{C}i = \mathrm{P}i \cdot \mathrm{Q}i + \mathrm{P}i \cdot \mathrm{R}i + \mathrm{Q}i \cdot \mathrm{R}i$. Explain the need for the block labelled "×2". For the particular value X7:0 = 5, determine the values of S13:0, C12:0 and T13:0. [8]

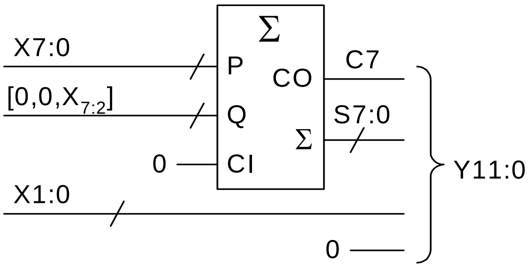


Figure 4.1

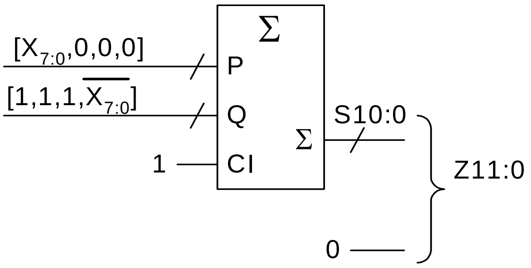


Figure 4.2

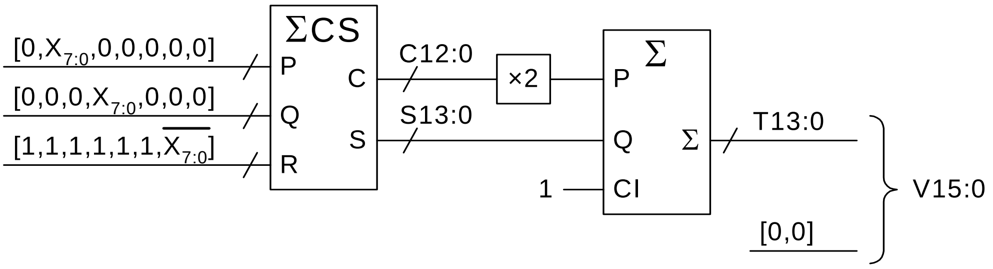


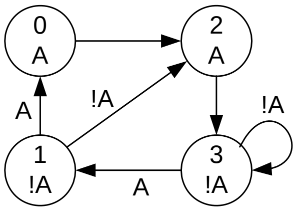
Figure 4.3

2005 E2.1/ISE2.2 Solutions

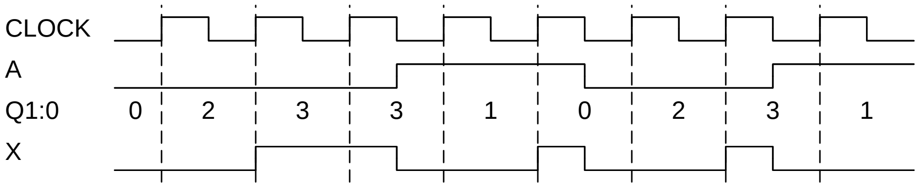
Key to letters on mark scheme: B=Bookwork, C=New computed example, A=Analysis of new circuit, D=design of new circuit

1. (a)

	D1:0/X	A=0	A=1
Q1:0	00	10/0	10/1
	01	10/1	00/0
	10	11/0	11/1
	11	11/1	01/0



[8A]



(b) The setup equation is:

$$t_p + t_g + t_s < ½T + t_g \quad \Rightarrow \quad 8 + 25 + 5 < ½T + 15 \quad \Rightarrow \quad T > 46$$

where the two t_g terms cannot be cancelled since they refer to different gates. The hold time equation is:

$$t_g + t_h < ½T + t_p + t_g \quad \Rightarrow \quad 25 + 2 < ½T + 8 + 15 \quad \Rightarrow \quad T > 8$$

Hence $T > 46\text{ ns} \quad \Rightarrow \quad f < 21.74\text{ MHz}$ [8C]

(c) The MSB is connected to the upper switch and so the switch positions correspond to $x = 110_2 = 6$.

The current through the MSB resistor is $\frac{1\text{ V}}{20\text{ k}\Omega} = 50\ \mu\text{A}$ which therefore corresponds to 4 LSB. It follows that $V = -30\text{ k}\Omega \times x \times ¼ \times 50\ \mu\text{A} = -0.375x\text{ V}$ [8C]

(d)

$$\mathrm{C0} = \mathrm{CG0} + \mathrm{CGP0} \cdot \mathrm{C} - 1$$

$$\mathrm{C1} = \mathrm{CG1} + \mathrm{CGP1} \cdot \mathrm{CG0} + \mathrm{CGP1} \cdot \mathrm{CGP0} \cdot \mathrm{C} - 1$$

[8B]

The expression for C1 should not be factorized because it would then entail three gate delays rather than only two (no marks deducted for factorization though).

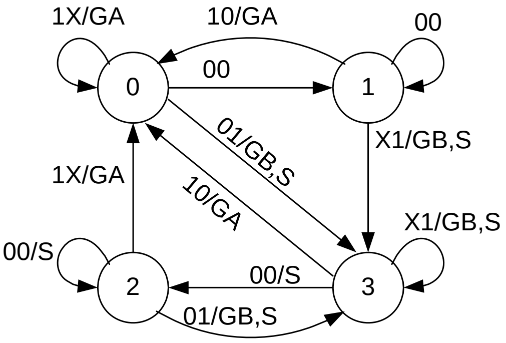
(e) The ROM will have 13 address inputs since 8192 = 2^{13}.

[8C]

$$\mathrm{CE} = \mathrm{CLOCK} \cdot \mathrm{READ} \cdot \mathrm{A15} \cdot \mathrm{A14} \cdot \overline{\mathrm{A13}}$$

2. (a)

		RA,RB			
NS1:0/GA,GB		00	01	11	10
S1:0	00	01/00	11/01	00/10	00/10
	01	01/00	11/01	11/01	00/10
	11	10/00	11/01	11/01	00/10
	10	10/00	11/01	00/10	00/10



Inputs RA,RB

Default Outputs:
GA= 0
GB=0
SELB=0 (denoted S)

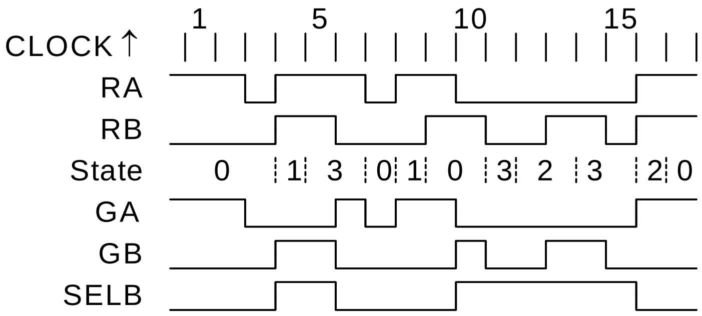
[12A]

No penalty for not showing SELB

Alternative output spec: States 0,2: /RA, !RA·RB, States 1,3: RA·!RB,RB

[12A]

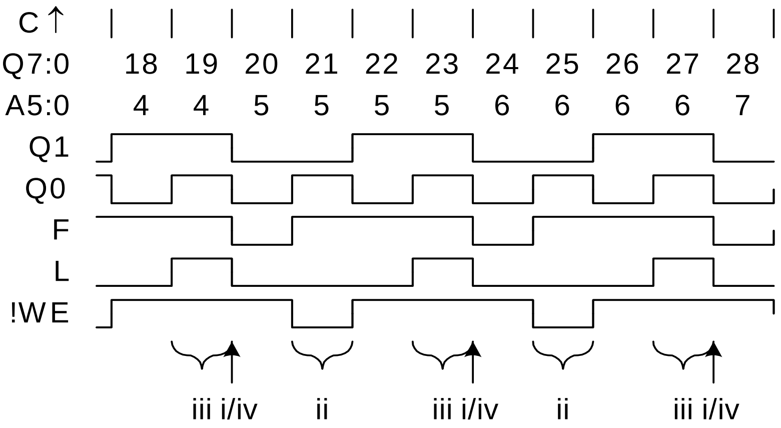
(b)



(c) (i) If both devices request access simultaneously, then access is granted by states 1 or 2 to the device that had it least recently. If GA was most recent you will be in state 1 while if GB was most recent you will be in state 2 and these states give priority to GB and GA respectively. Thus from state 1, you will branch to 3 whenever RB is high but to state 0 only if RA is high <u>and</u> RB is low. [3A]

(ii) If one device requests access while the other is in the middle of transmitting data, then it has to wait for access until the active device lowers its request line. Thus for example if A is transmitting you will be in state 0 and, once there, you will remain as long as RA stays high. When RA goes low, you will branch to 1 or 3 according to whether RB is high or not. [3A]

3.. (a),(b)



[14A]

[6A]

X7:0 is stored on the rising edge marked “i” since L is high. It is then stored in address location 5 when !WE goes low at “ii”. Since N5:0 = 0, the RAM address is still equal to 5 at “iii” when the value is read out of RAM and stored in the output register at “iv”.

[10A]

(c) (i) When N5:0=0 as in the diagram, X7:0 will change on the rising edge of Q1 at the start of cycle 18 and be output at the start of cycle 24 as described above. This gives a delay of 6 clock cycles.

(ii) When N5:0 = 63 = –1 (modulo 64), the value of A5:0 will be reduced by 1 when L goes high. This means that in the above diagram, the value of X7:0 that was written in cycle 21 will not be retrieved until cycle 25. This will add another 4 clock cycles onto the delay with will now equal 10 clock cycles.

(iii) We can deduce a general formula from the previous two answers:

$$\text{delay} = 6 + 4(64 - n)_{\text{mod } 64}$$

This correctly gives a minimum delay of 6 when *n*=0 and increments by 4 cycles if *n* is decreased by 1. It gives a delay of 258 clock cycles when *n* = 1.

[6A]

4. (a) The number 10 is 1010_2. Therefore to multiply by 10, we need to perform the addition Y = 8*X + 2*X:

	X7	X6	X5	X4	X3	X2	X1	X0	0	0	0
			X7	X6	X5	X4	X3	X2	X1	X0	0
Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

We see that Y0=0 always and that Y2:1 = X1:0. We therefore need to add [X7:0] onto [0, 0, X7:2]. Since these are unsigned numbers, we can use the C7 output from the adder to form the MSB of the output.

An alternative, more algebraic, approach is to say that $P=X_{7:0}=4*X_{7:2}+X_{1:0}$. It follows that $[C_7, S_{7:0}] = 5*X_{7:2} + X_{1:0}$. But $Y_{11:0} = 8*[C_7, S_{7:0}] + 2*X_{1:0} = 40*X_{7:2} + 8*X_{1:0} + 2*X_{1:0} = 10*(4*X_{7:2} + X_{1:0}) = 10*X_{7:0}$.

[7A]

(b) For this circuit, we have:

X7	X6	X5	X4	X3	X2	X1	X0	0	0	0	0
1	1	1	!X7	!X6	!X5	!X4	!X3	!X2	!X1	!X0	0
										1	0
Z11	Z10	Z9	Z8	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0

Thus we are calculating [Z11:0] = (16–2) × [X7:0] =14 × [X7:0]

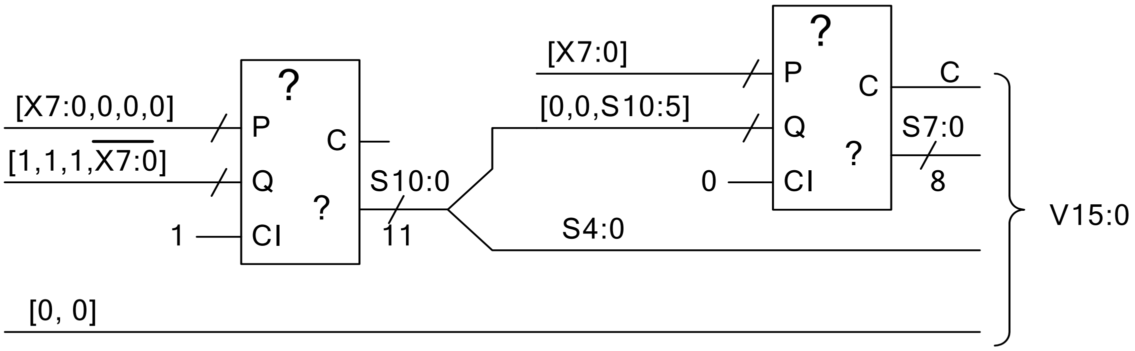
Note that since we are adding a negative number onto a positive one, there is no need to use the C10 bit at all.

We have $0 \le S10:0 \le 255 \times 7 = 1785 = 11011111001_2$

[9D]

(c) We need to calculate (128+32–4) × [X7:0]

	X7	X6	X5	X4	X3	X2	X1	X0	0	0	0	0	0	0	0
			X7	X6	X5	X4	X3	X2	X1	X0	0	0	1	0	0
1	1	1	1	1	1	!X7	!X6	!X5	!X4	!X3	!X2	!X1	!X0	0	0
V15	V14	V13	V12	V11	V10	V9	V8	V7	V6	V5	V4	V3	V2	V1	V0



(d) The carry save adder converts the sum of three numbers (P, Q, R) into the sum of two numbers (C and S). The carry output from each stage has a value equal to twice that of the sum output and so we need to multiply it by 2 before adding it onto the sum. The multiplication does not require any logic circuitry. [8A]

With the specific value [X7:0] = 5, we have

	0	0	0	0	0	1	0	1	0	0	0	0	0	P13:0
			0	0	0	0	0	1	0	1	0	0	0	Q13:0
1	1	1	1	1	1	1	1	1	1	1	0	1	0	R13:0
1	1	1	1	1	1	0	1	1	1	0	0	1	0	S13:0
	0	0	0	0	0	1	0	1	0	1	0	0	0	C12:0

P13:0 = 160, Q13:0 = 40, R13:0 = –6

S13:0=11 1111 0111 0010=16242=–142

C12:0=0 0000 1010 1000=168

T13:0 = 2×168 – 142 + 1 = 195 = (5×156)/4 = 1100 0011