E2.1

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2003

DIGITAL ELECTRONICS 2

Friday, 13 June 2:00 pm

Time allowed: 2:00 hours

There are FIVE questions on this paper.

Answer THREE questions.

Corrected Copy

Corrected Copy

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

Examiners responsible	First Marker(s) :	D.M. Brookes
	Second Marker(s) :	T.J.W. Clarke

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 synchronous state machine is required having two inputs, P and Q, and two outputs, X and Y. All signal transitions occur shortly after the CLOCK rising edge. The signals P and Q contain high-level pulses each lasting for exactly one clock cycle and never remain high for more than one consecutive clock cycle.

   Figure 1.1 shows typical waveforms for P and Q and the corresponding output waveforms X and Y. Output X should go high whenever two or more pulses occur on P without an intervening pulse on Q. Similarly, output Y should go high whenever two or more pulses arrive on Q without an intervening pulse on P. If pulses arrive alternately on the two inputs, then both X and Y should remain low.

   (a) For the input signals shown in *Figure 1.1*, complete a state sequence that is consistent with the above specification. The states should be labelled "a", "b", ... and have been already completed for the first few CLOCK cycles. [10]

   (b) Assuming that P and Q are never high simultaneously, draw a state diagram for the circuit indicating clearly the state transitions for all other input combinations and defining the values of X and Y in each state. [10]

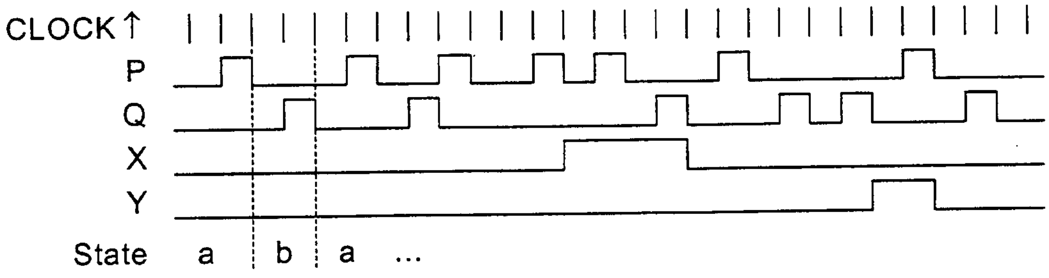


Figure 1.1

2. In the counter circuit of *Figure 2.1* one of the flipflops includes an asynchronous reset input which forces Q2 to zero whenever RST=1.

   (a) For the case RST=0, draw the state diagram for the circuit of *Figure 2.1* and draw a timing diagram showing the waveforms of the clock, CQ, and the outputs Q0, Q1 and Q2 for one complete cycle of the counter. [7]

   (b) Repeat part (a) for the case RST=1. [5]

   (c) Draw the state diagram for the counter circuit of *Figure 2.2*. Draw a timing diagram showing, for the case MOD=0, the waveforms of the clock, CS, and the outputs S0, S1 and Z for one complete cycle of the counter. [4]

   (d) The two circuits are combined to form a single counter by connecting CS to Q0 and RST to Z. The signal MOD is used as a control input and the clock input to the combined circuit is CQ. Giving your reasons fully, determine the number of CQ cycles between consecutive rising edges of S1 when (i) MOD=1 and (ii) MOD=0. [4]

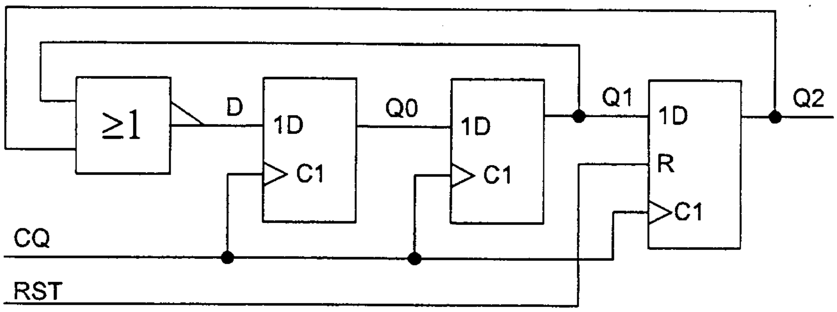


Figure 2.1

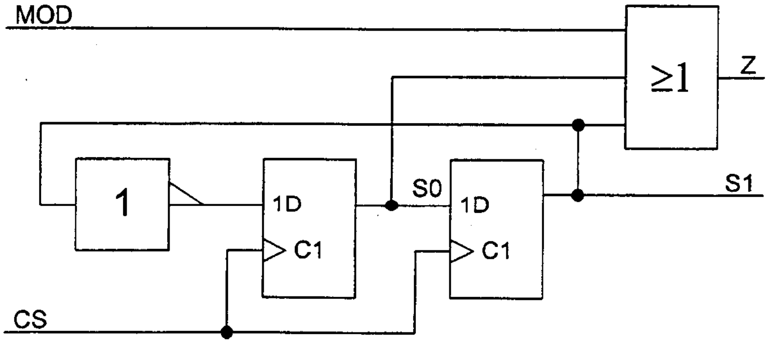


Figure 2.2

3. The circuit of *Figure 3.1* shows a microprocessor connected to three memory circuits: two 8k×8bit read/write memories (RAMs) and one 16k×8bit read-only memory (ROM). The A, D, WE and OE signals are connected to multiple memory circuits as indicated in the Figure. The signal CLOCK has period T and unity mark:space ratio.

(a) The circuit module marked "Decode Logic" in *Figure 3.1* generates the chip-enable signals for the memory circuits. Design the module so that E0, E1 and E2 are true if and only if A15:0 lies in the hexadecimal address ranges \$0000-\$1FFF, \$2000-\$3FFF and \$B000-\$EFFF respectively. You may use only AND, NAND, OR, NOR gates and Inverters. [7]

(b) *Figure 3.2* shows the timing specifications for the microprocessor when it is reading from memory. Data is read into the microprocessor 1½T after the start of the cycle with a setup time of t_s. The propagation delays of the ROM from its address, chip enable and output enable inputs are t_A, t_C and t_O respectively. All logic gates have a propagation delay of t_g. [9]

Write down all the timing inequalities that must be satisfied in order to meet the setup time requirement t_s when reading data from the ROM.

(c) Determine the smallest clock period, T, for which the t_s requirement is satisfied if the timing parameters in ns are: t_g=5, t_d=40, t_m=30, t_s=20, t_A=70, t_O=20, t_C=60 [4]

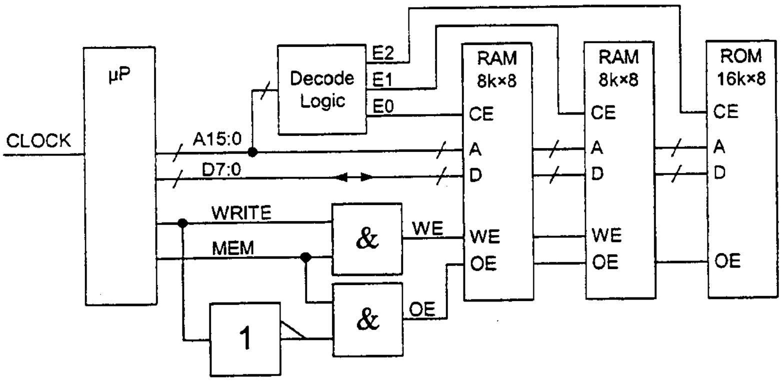


Figure 3.1

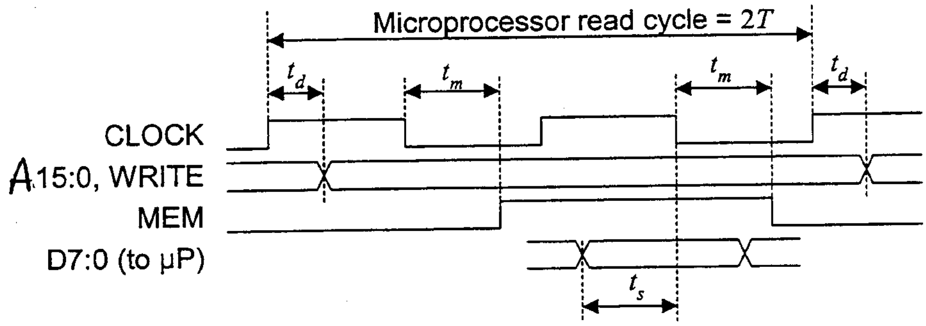


Figure 3.2

4. *Figure 4.1* shows the carry logic for stage n of a full adder circuit. Eight identical stages are cascaded to form the carry logic for an 8-bit adder with $n = 0$ to 7.

(a) Determine the worst case propagation delay in the 8-bit adder from C–1 to C7. Each gate has a delay of 1 unit. [2]

(b) Show that the truth table of the circuit of *Figure 4.1* remains unchanged if inverters are placed at the output and at each of the three inputs. [3]

(c) Show how, by inverting the inputs and outputs of odd-numbered stages only, it is possible to reduce the propagation delay of the 8-bit adder's carry logic by merging gates between adjacent stages and by representing the intermediate carry signals as "bundles", i.e. a group of signals whose AND has the required value. Illustrate your answer by showing the resultant circuitry for the stages n = 2, 3 and 4. Show what additional gates are needed to generate the signals C2 and C3 explicitly from the corresponding bundles. [5]

(d) Determine, for your circuit of part (c), the worst case propagation delay from C–1 to the C7 bundle. [1]

(e) In *Figure 4.2*, the 8-bit adder is followed by a carry-skip multiplexer. The signal C7X has the same value as C7 but has a lower propagation delay under some circumstances. Implement the multiplexer using logic gates to generate the carry signal, !C7X, in bundle form with only one gate delay in the data path. Give a Boolean expression for SEL, the select input of the multiplexer. [4]

(f) Calculate, for your circuit of part (e), the worst case propagation delay to the !C7X bundle from (i) C–1, (ii) P0 and (iii) P1. [5]

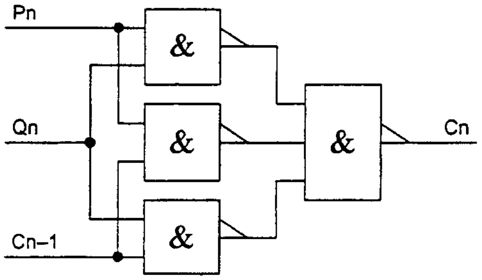


Figure 4.1

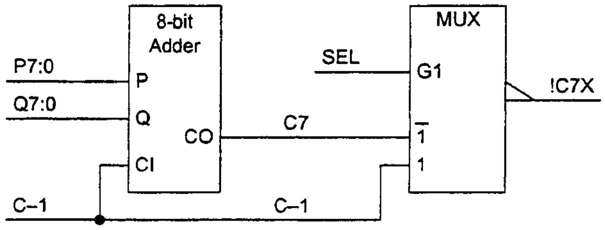


Figure 4.2

5. *Figure 5.1* shows part of the circuit for a Digital-to-Analog converter whose output voltage is proportional to the value of its 3-bit twos-complement input number X2:0. The circuit includes a module that generates four equal current-source outputs of value I_0, and a current mirror whose output current is equal in magnitude but opposite in sign to its input current. The Boolean signals A, B, C and D take the values 0 or 1 and are used to control switches which are shown in *Figure 5.1* in the positions corresponding to A=B=C=D=0.

(a) Explain why I_y = (B+C+D)I_0 and derive a similar expression for I_x in terms of A, B, C and D. [3]

(b) Give an expression for V_{out} as a function of A, B, C and D. Determine the minimum and maximum values of V_{out} and the combinations of A, B, C and D that give rise to them. [3]

(c) Show that it is possible to generate sequentially at V_{out} all integer multiples of I_0R from $-4I_0R$ to $+3I_0R$ in such a way that at most two of A, B, C and D change between any pair of successive values. [5]

(d) Using standard logic gates, design the circuitry needed to generate A, B, C and D from the input signals X2, X1 and X0 so that the condition of part (c) is met. [4]

(e) If each of the current source outputs has a maximum error of ±1% and the current mirror has a gain error of up to ±1%, determine (i) the maximum voltage error in Vout and (ii) the maximum voltage error in the amount that V_{out} changes when X2:0 is incremented by unity. In each case give a value of X2:1 for which the worst-case error occurs. [5]

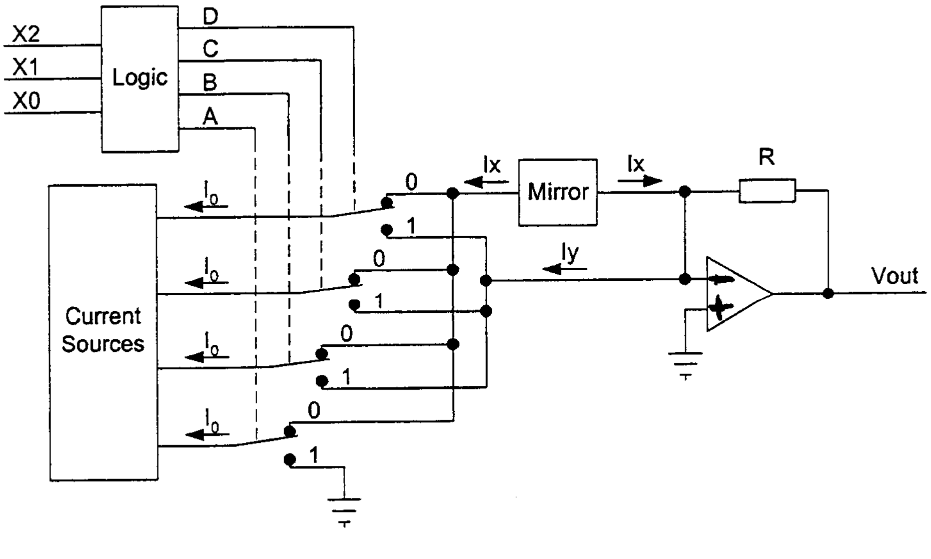
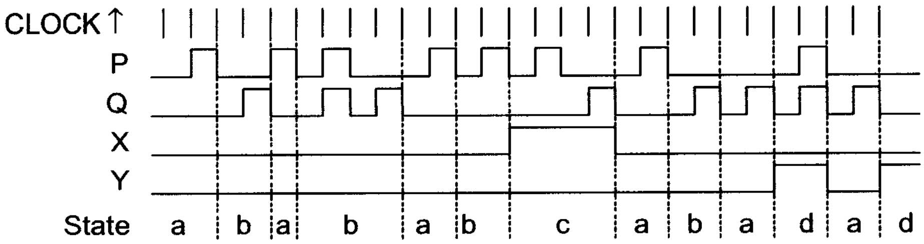


Figure 5.1

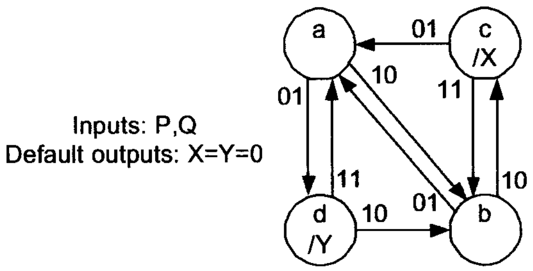
2003 E2.1/ISE2.2 Solutions

1. (a)



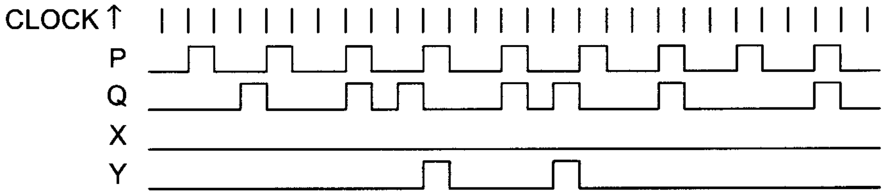
[10]

(b) In all cases we remain in the current state if P=Q=0.



[5]

(c) In the figure, P and Q have almost the same frequency by a small amount of jitter means that some pulses from Q arrive slightly late. Near the end of the diagram, a Q pulse is missing.

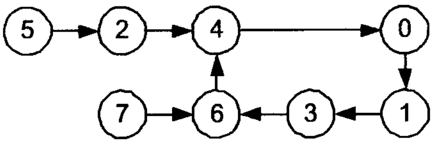


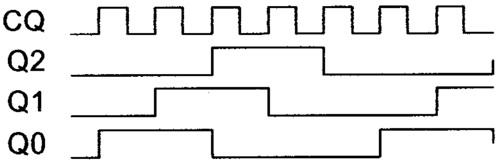
The original state machine specification will treat the order of arrival as PQPQPQPQPQPQPPQP. It will not give an error until the penultimate P pulse

The revised design, on the other hand, will treat the order of arrival as PQPPQQPPQPPQPPQ. The repeated P pulses will not give rise to errors since they are immediately cancelled by a Q. The repeated Q pulses will however give errors as shown above. Even worse than these spurious errors, the revised design will not notice any errors in the pattern at the end of the diagram where Q is in fact half the frequency of P. [5]

2. (a)

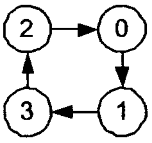
Q2	Q1	Q0	Q1	Q0	D
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	1	1	0

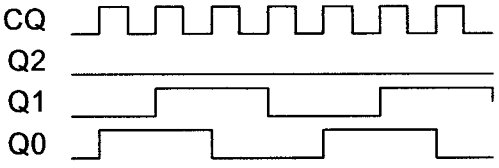




[7]

(b) Q2 is always zero so we get a cycle of 4 rather than 5:

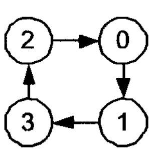


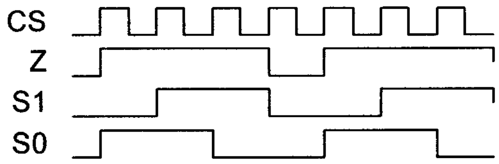


[5]

(c) This is in fact exactly the same circuit as part (b). Provided that MOD=0, Z goes low when the counter is in state 0. If MOD=1, then Z never goes low.

[4]





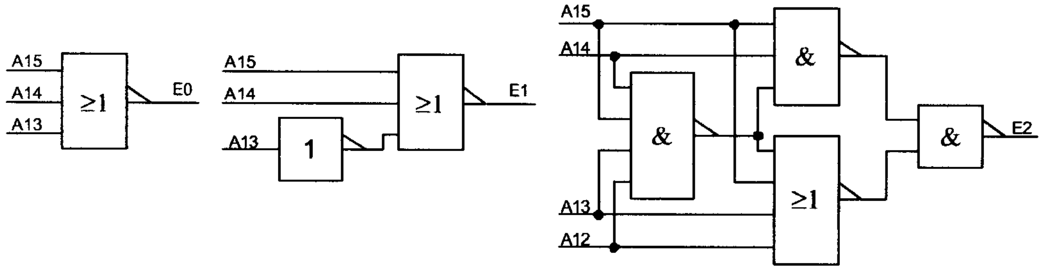
(d) (i) If MOD=1 then Z is always high and the Q2 flipflop is permanently reset. Rising edges of Q0=CS therefore occur every four CQ cycles and, since it takes four of these for a complete cycle of the S counter, there will be 16 CQ cycles between successive S1 rising edges.

(ii) If MOD=0 then Z will go low for one in four cycles of the Q counter and when this happens, the cycle length will increase from four CQ clocks to five. There will therefore be 17 CQ cycles between successive S1 rising edges. [4]

3. (a) We need:

E0=!A15·!A14·!A13, E1=!A15·!A14·A13, E2= (A15·A14)⊕(A15·A13·A12)

E2 may be implemented in many different ways, with or without an XOR gate.



[6]

(b) The ROM needs 14 address inputs, so A13:0 will be connected to the address inputs. Because \$B000 is not a multiple of 2^{14} = \$4000, the values will be stored in the ROM with an address offset. Starting at address 0 in the ROM, we will have microprocessor addresses \$C000 to \$EFFF followed by \$B000 to \$BFFF. [3]

(c) There are four paths to consider:

Path	Inequality	Numerical
A15:0 → E2 → CE → D	$t_d+3t_g+t_C+t_s < 1\frac{1}{2}T$	$1\frac{1}{2}T > 135$
A15:0 → A → D	$t_d+t_A+t_s < 1\frac{1}{2}T$	$1\frac{1}{2}T > 130$
WRITE → OE → D	$t_d+2t_g+t_O+t_s < 1\frac{1}{2}T$	$1\frac{1}{2}T > 90$
MEM → OE → D	$t_m+t_g+t_O+t_s < T$	$T > 75$

[8]

(d) From the numerical calculations in the table above, we find that the critical path is the one through the decode logic and that T>90. [3]

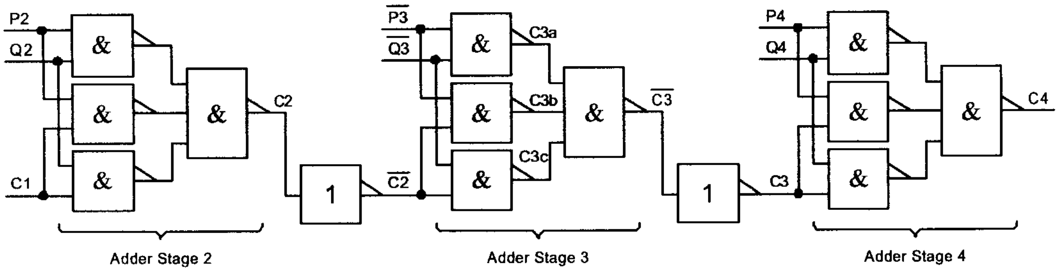
4. (a) Each stage has a delay of 2 units so the total delay is 16 units. [2]

(b) Cn is true iff either two or all three of the inputs are true.

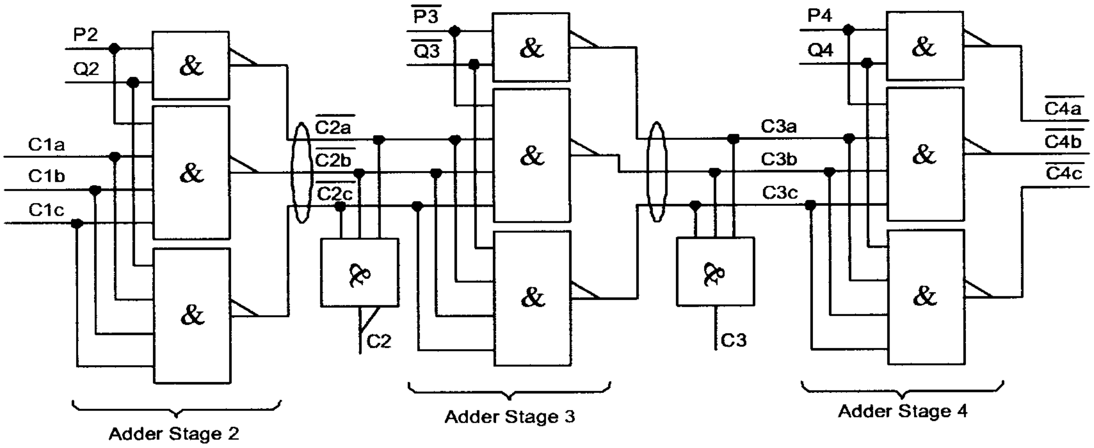
Cn is **false** iff either zero of one of the inputs are true; this is the same as saying that either two or all three of the inputs are **false**. Thus !Cn is true iff either two or three of the inverted inputs are true. QED. [3]

Alternatively the proposition can be proved directly from the truth table.

(c) The diagram shows stages 2, 3 and 4 with inverters either side of stage 3. [6]



The shaded gates can be merged to form two 4-input NAND gates:



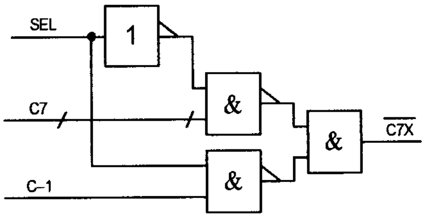
Even numbered carry bundles are inverted while odd numbered ones are not. The explicit generation of the carry outputs therefore requires a NAND or AND gate for even and odd numbered stages respectively. These are shown above with shaded backgrounds.

(d) The worst case propagation delay is now only 8 gate delays. [2]

(e) When SEL=1, it will select C-1 to be connected to its output. This must happen only when all bit positions of the adder are propagating the carry, i.e.

SEL = P0⊕Q0 • P1⊕Q1 • ... • P7⊕Q7 [4]

An inverting multiplexer has the following circuit:



where the C7 input is in fact a bundle. We may omit the final AND gate if we permit !C7X to be in bundle form. This leaves only one gate delay in the data path.

(f) (i) The worst case delay from C-1 to !C7X is only one gate delay. This is the entire point of carry-skip. [3]

(ii) The worst case delay from P0 to !C7X is 8 within the adder plus one additional gate delay in the MUX to give a total of 9 gate delays.

(iii) The worst case delay from P1 to !C7X is also 9 gate delays. Although it doesn't have to go throughthe first stage, there is an additional inverter delay needed to generate !P1.

5. (a) I_y comes from the three switches that are controlled by B, C and D. Each switch controls a current of I_0, so the total current is $I_y = (B + C + D)I_0$ as stated in the question.

I_x is connected to the "0" position of all four switches and so its current is $I_x = (4 - A - B - C - D)I_0$ [3]

(b) $V_{out} = (I_y - I_x)R = (A + 2B + 2C + 2D - 4)I_0R$

The minimum, $V_{out} = -4I_0R$, is when A=B=C=D=0. [3]

The maximum, $V_{out} = +3I_0R$, is when A=B=C=D=1.

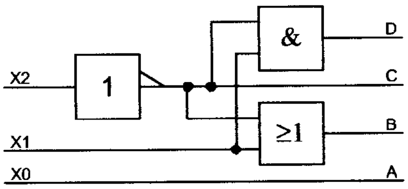
(c) The basic idea is to use A to choose whether V_{out} is an even or odd multiple of I_0R and then to add in B, C and D one at a time for the rest.

X2 X1 X0	A B C D	$V_{out}(I_0R)^{-1}$
1 0 0	0 0 0 0	-4
1 0 1	1 0 0 0	-3
1 1 0	0 1 0 0	-2
1 1 1	1 1 0 0	-1
0 0 0	0 1 1 0	0
0 0 1	1 1 1 0	1
0 1 0	0 1 1 1	2
0 1 1	1 1 1 1	3

[5]

It may be seen that between any two adjacent rows, only A and at most one of B, C and D changes state.

(d) From inspection, A=X0, B=!X2+X1, C=!X2, D=!X2•X1



[4]

(e) (i) The currents that contribute to Ix have the largest errors since they have the additional error of the current mirror. The largest absolute error is therefore when X2:0=-4 and both the current sources and the mirror gain are 1.01 times their correct value. The error is then $-4(1.01^2 - 1)I_0R = -0.0804I_0R$.

(ii) When two switches change (e.g. -3 to -2, -1 to 0 or 1 to 2) the change in V_{out} due to switch A is between $-0.99I_0R$ and $-1.01I_0R$.

[5]

The change in V_{out} due to switch B, C or D is between $0.99(1+0.99)I_0R = 1.9701I_0R$ and $1.01(1+1.01)I_0R = 2.0301I_0R$.

The total change is therefore between $0.9601I_0R$ and $1.0401I_0R$ and the worst case error is $0.0401I_0R$.