Paper Number(s): **E2.1**
ISE2.2

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE
UNIVERSITY OF LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2000

EEE/ISE PART II: M.Eng., B.Eng. and ACGI

DIGITAL ELECTRONICS 2

Monday, 12 June 2000, 2:00 pm

There are FIVE questions on this paper.

Answer THREE questions.

All questions carry equal marks.

Time allowed: 2:00 hours

questions before 3pm: **Corrected Copy**

Q2; Fig2 : Assume the Logic block is non-inverting

Q5 c) : x is the value of the one-bit number X

after 3pm:
3:55 Q5 lines : "OR-gate" should read "inverter"

Examiners: Mr D.M. Brookes, Dr 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 least significant bit of a binary number is always designated bit 0. Signals labelled with identical names are connected together. Logical negation is indicated by an exclamation mark: i.e. !X denotes the logical inversion of X.*

1. Figure 1 shows the circuit of a synchronous state machine having a single input signal, X and a state defined by the 2-bit binary number S0:1.

   (a) Construct the state table for the circuit so that the state S0:1 follows the sequence 0, 1, 2, 3, 0, ... when X=0 and follows the sequence 0, 1, 3, 2, 0, ...when X=1. Draw a state diagram for the circuit. [8]

   (b) Derive simplified Boolean equations that express D0 and D1 in terms of X and S0:1. You are not required to implement these equations using gates. [4]

   (c) Complete the timing diagram of *Figure 1* by showing the sequence of states followed by the circuit. The circuit is initially in state 0 as shown in the diagram. [4]

   (d) If X changes slightly before the CLOCK rising edge, it is possible for the state machine to follow an incorrect state sequence. Giving reasons for your answer, identify which incorrect state transitions are possible. Explain how the circuit may be modified in order to prevent such transitions. [4]

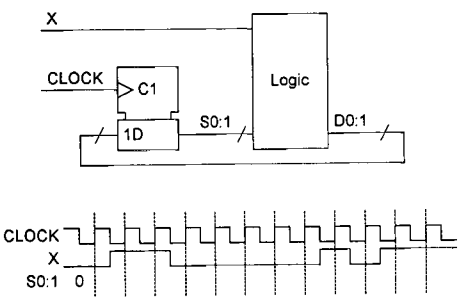


Figure 1

2. *Figure 2* shows a circuit containing three flipflops and a logic block. As indicated on the diagram, flipflop [B] is separated from the other two by a long cable with a propagation delay of P.

The logic block has a propagation delay of 7 ns and the flipflop timing parameters are: setup=2 ns, hold=3 ns, propagation delay=5 ns. The signal CLOCK is a symmetrical squarewave with a period of T. The input signal U changes on the falling edge of the CLOCK as shown.

(a) If P=30 ns and the CLOCK period is 100 ns, complete the timing diagram shown in the figure by drawing the waveforms of V, W, X, Y and Z. Calculate the time delay to the rising edges of V, W, X, Y and Z from the CLOCK rising edge that follows the rising edge of U. [6]

(b) If P may vary independently for each wire in the range 20 ns to 30 ns, explain why the circuit may fail to meet the hold-time requirements of flipflop [B]. [4]

(c) Flipflip [B] is replaced by a flipflop having identical timing parameters but that responds to its <u>falling</u> CLOCK edge. Write down the setup and hold inequalities for both flipflop [B] and flipflop [C]. Hence determine the maximum clock frequency for reliable operation of the circuit when P may vary independently for each wire in the range 20 ns to 30 ns. [10]

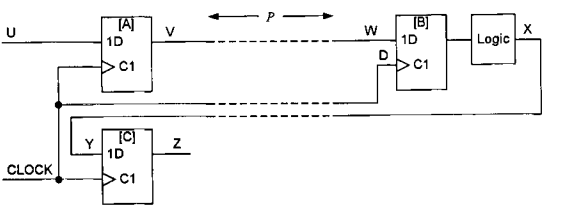


Figure 2

3. (a) *Figure 3* shows a circuit consisting of a 3-bit adder and three inverters. The variables *p, q, y* and *z* denote the values of P0:2, Q0:2, etc. as unsigned binary numbers. Give an arithmetic expression for *p* in terms of *y* and hence show that [5]

$$z = \begin{cases} q - y & \text{for } q \geq y \\ q - y + 8 & \text{for } q < y \end{cases}$$

(b) The circuit of *Figure 4* comprises a register, two counters and the subtractor circuit of part (a), labelled Δ in the diagram. The 3-bit counter increments on the rising edge of CLOCK provided that X=1. The register is loaded on the rising edge of CLOCK whenever K0:1=3. Complete the timing diagram by showing the values taken by *k, q, y* and *z* during each CLOCK pulse. You may assume that all counters and registers are initially reset. [10]

(c) Explain the relationship between *z* and the values of *x* at previous CLOCK rising edges and determine the maximum value that *z* can take. [5]

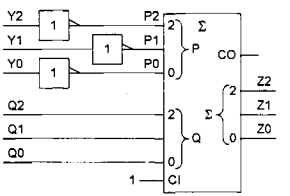


Figure 3

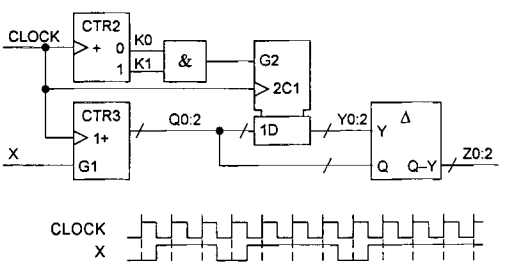


Figure 4

4. *Figure 5* shows the circuit of a 3-bit carry-lookahead adder. Each full-adder module has two carry outputs: !CG = !(P·Q) and !CGP = !(P+Q).

(a) Explain why the signals CG and CGP may be called "Carry Generate" and "Carry Generate/Propagate" respectively. [3]

(b) Give a simplified Boolean expression for C1 in terms of !CIN, !CG0, !CGP0, !CG1 and !CGP1. [4]

(c) Show the circuit diagram for a complex CMOS gate that implements the Boolean expression of part (b). [7]

(d) The propagation delay of the logic blocks that generate the C*i* signals is 1 unit and the propagation delays of the adder modules are:

From	To	Delay
P,Q,CI	S	3 units
P,Q	!CG, !CGP	1+0.2*k* units

where *k* denotes the number of logic blocks to which the !CG or !CGP output is connected.

Determine the worst-case delay from any P or Q input to any S output of the 3-bit adder. Do not consider any additional loading from the connections marked "to later stages" on the diagram. [3]

(e) Determine the worst-case delay from any P or Q input to any S output for an *n*-bit carry-lookahead adder that is constructed in the same way. You should include the effects of additional loading from the connections marked "to later stages" on the diagram. [3]

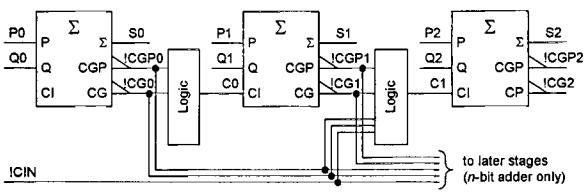


Figure 5

5. *Figure 6* shows the circuit of a 3-bit successive approximation Analog-to-Digital converter designed for input voltages in the range $-4.5 < V < 3.5$ volts. The circuit consists of a comparator, a Digital-to-Analog converter (DAC), an or-gate, a register and a logic block. On the rising edge of the CLOCK, the register is loaded with new data if START=DONE=0 and is reset if START=1.The comparator output, HI, is true whenever the input voltage, V, is greater than the DAC output, X.

A conversion is initiated by taking START high for one clock cycle to reset the register. At the end of a conversion, the value of Q0:2 represents the input voltage V rounded to the nearest number of volts and expressed as a signed 2's-complement number. Thus an input of -1.9 volts will be converted to Q0:2 = -2 (i.e. 110 in binary).

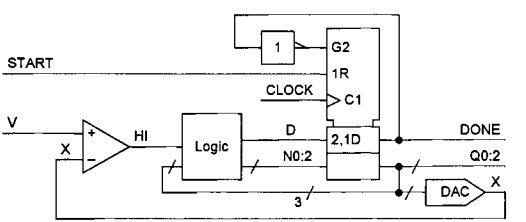
The register and logic block form a synchronous state machine whose state is defined by the signed 3-bit number Q0:2. *Figure 6* shows an outline state diagram indicating the possible transitions when START=DONE=0 from which the state transition conditions have been omitted.

(a) Explain why the sequence of states that should be followed when converting an input voltage, V, of -1.9 volts is 0, -2, -1, -2. [3]

(b) Draw a timing diagram showing the process of conversion when $V = -1.9$ volts. Your diagram should show the waveforms of CLOCK, START, HI, D and DONE and give the values of X and Q0:2. [8]

(c) Give a general expression in terms of the state, n, for the DAC output voltage, X. [4]

(d) Assuming the conditions START=DONE=0, indicate the value of D in each state of the state diagram and the value of HI for which each transition should be taken. [5]



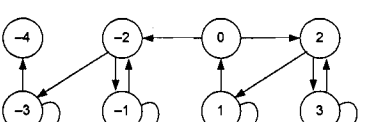


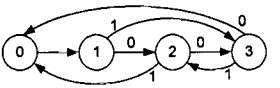
Figure 6

2000 E2.1/ISE2.2 Solutions

1. (a) From the specification in the question we get the following state table:

D1,D0	X	
S1,S0	0	1
00	01	01
01	10	11
11	00	10
10	11	00

[4]



[4]

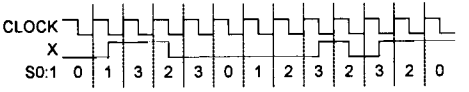
(b) $D1 = \overline{X}(S1 \oplus S0) + X \cdot S0 = (X + \overline{S1}) \cdot S0 + \overline{X} \cdot S1 \cdot \overline{S0}$

$D0 = \overline{X} \cdot \overline{S0} + X \cdot \overline{S1} = \overline{(X + S0)} + X \cdot \overline{S1}$

[4]

Either of the expressions for D0 and D1 are OK.

(c)



[4]

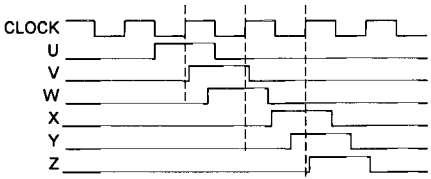
(d) The problem can only arise when the two alternative next states differ in more than one bit posiion. This only happens in state 2 when the alternative next states are 0 and 3.

If X changes just before the clock rising edge, then D0 and D1 may change during the setup/hold window of the register and the circuit may branch to any one of the four states. It is an error if it branches to either 1 or 2. [4]

The problem may be prevented by passing the signal X through a register. This delays it by one clock cycle but ensures that it is synchronised to the clock.

D. M. Bull T. Clarke

2. (a)



[3]

[3]

The time delays to V, W, X, Y and Z are respectively 5, 35, 142, 172 and 205 ns.

(b) The latest that D can change is 30 ns after the CLOCK rising edge and so W should not change until 3 ns after this. However, the earliest that W can change is actually 25 ns after the CLOCK rising edge which is 8 ns too early. If, for example the clock delay is 30 ns and the data delay is between 25 and 28 ns, then W will change during the hold interval. [4]

(c) Let $f = 5$ ns, the flipflop delay, $P = 20 - 30$ ns, the wire delay, $l = 7$ ns, the logic delay, $s = 2$ ns, setup and $h = 3$ ns, hold.

[B], setup: $f + P_{max} + s < \frac{1}{2}T + P_{min} \Rightarrow \frac{1}{2}T > 5+30+2-20 = 17$ ns

[B], hold: $P_{max} + h < \frac{1}{2}T + f + P_{min} \Rightarrow \frac{1}{2}T > 30+3-5-20 = 8$ ns [8]

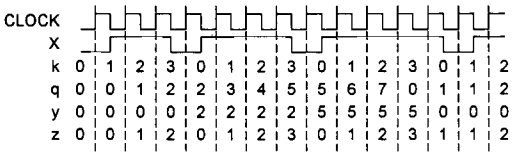
[C], setup: $P_{max} + f + l + P_{max} + s < \frac{1}{2}T \Rightarrow \frac{1}{2}T > 30+5+7+30+2 = 74$ ns

[C], hold: $h < \frac{1}{2}T + P_{min} + f + l + P_{min} \Rightarrow \frac{1}{2}T > 3-20-5-7-20 = -49$ ns

Hence $\frac{1}{2}T > 74$ ns giving a maximum frequency of 1/148 GHz = 6.76 MHz. [2]

3. (a) We have $p = 7 - y$. Hence, since the carry in is high, the adder calculates $d = q - y + 8$. However since we are only taking the low order 3 bits of the answer we will subtract 8 from this answer whenever it is ≥8. The required answer follows. [5]

(b)



k	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2
q	0	0	1	2	2	3	4	5	5	6	7	0	1	1	2
y	0	0	0	0	2	2	2	2	5	5	5	5	0	0	0
z	0	0	1	2	0	1	2	3	0	1	2	3	1	1	2

[10]

(c) z counts the number of clock rising edges for which X has been high counting from the most recent clock edge that caused k to reset to 0. It follows that z can take values in the range 0 to 4. [5]

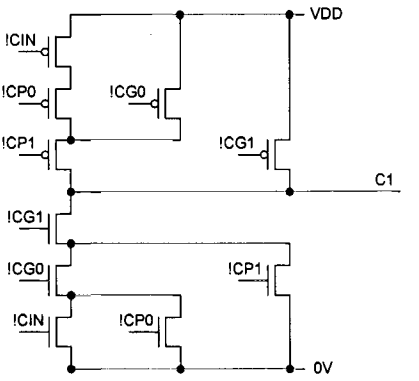
4. (a) An adder stage generates a carry into the next column if 2 or more of its inputs are high. If P·Q=1 the adder stage will generate a carry out into the next column regardless of the level of its own carry in signal. CG detects this case.

If P⊕Q=1 then at exactly one of P and Q is high. It follows that the stage will generate a carry if and only if the carry input is also high: this is called propagating the carry. CGP=1 if either P or Q is high: this includes both the carry generate and carry propagate cases. [3]

(b) $C1 = CG1 + CP1 \cdot (CG0 + CP0 \cdot CIN) = \overline{(\overline{CG1} \cdot (\overline{CP1} + \overline{CG0} \cdot (\overline{CP0} + \overline{CIN})))}$ [4]

We need it in this form for the next part.

(c)



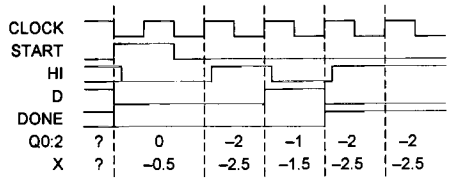
[7]

(d) The worst-case delay is P0 → !CG0 → C_i → S_{i+1}. This delay is 1.4+1+3 = 5.4 units. Note that the delays via !CG0 and !CGP0 are identical. [3]

(e) For an n-bit adder, the delay is the same as before but the delay to the !CG0 output is now $1+0.2(n-1)$ giving a total delay of $4.8+0.2n$. The delay will be greater than this for large n because of increased wiring capacitance and the logic blocks will need to be composed of multiple stages rather than being implemented as a single complex gate. [3]

5. (a) The sequence of comparisons must be –0.5, –2.5, –1.5. The final comparison selects between a final state of –2 and –1 for which –2 is the correct decision. Thus the sequence of states must be 0, -2, -1 –2. During the final state, DONE=0 and so the circuit freezes until ST goes high again to initiate another conversion. [3]

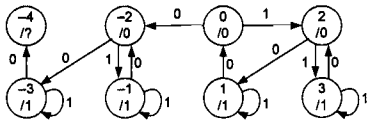
(b)



[8]

(c) In state n we must decide between a value of n and a value of n–1. The threshold between these is n–0.5 volts and so this is the value that X must take. [4]

(d)



I/O signals: ST/D

[5]