2013 – Part IIA Module 3B2 – Integrated Digital Electronics

1. (a) The logic swing is 1.2V, symmetric about $V_R = -1.5$v.

$$V_{OH} = V_R + \frac{1}{2}\times 1.2 \quad = \quad -1.5 + 0.6 \quad = \quad -0.9\ \text{V}$$

$$V_{OL} = V_R - \frac{1}{2}\times 1.2 \quad = \quad -1.5 - 0.6 \quad = \quad -2.1\ \text{V}$$

and $V_{OH} - V_{OL} = 1.2$ V as required. In this design, V_{O1} is sometimes referred to as the 'NOR' output; V_{O2} is the 'OR' output; they are always mutually inverted.

As a matter of principle, the maximum allowable I_E should be used to maximise switching speed, so in the following we take $I_E = 5$ mA, wherever possible. When input A is **high** (–0.9 V),

$V_{E1} = V_{OH} - V_{BE1} - V_{EE}$ and

$R_E = (V_{OH} - V_{BE1} - V_{EE}) / I_{E1} \quad = \quad (-0.9 - 0.8 + 6) / 5 \times 10^{-3} = 860$ ohms

We need to choose R_{C1} so that V_{O1} is at $V_{OL} = -2.1$ V when T_3 is ON, and $I_E = 5$ mA

$R_{C1} = V_{CC} - (V_{CC} + V_{BE3}) / I_{E1} \quad = \quad (0 + 2.1 - 0.8) / 5 \times 10^{-3} = 260$ ohms

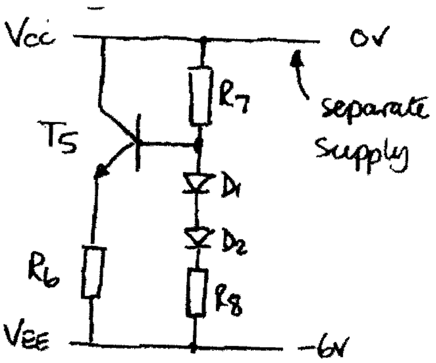
R_{C2} is not just the same as R_{C1}, since V_R is held fixed at –1.5 V. We choose its value so that V_{O2} is at $V_{OL} = -2.1$ V when A is **low**. Note that its emeitter current I_{E2} will not be exactly 5mA, since the value of R_E has already been fixed.

$I_{E2} = (V_R - V_{BE2} - V_{EE}) / R_E \quad = \quad (-1.5 - 0.8 + 6) / 860 = 3.7/0.86 = 4.2$mA

which is close to 5mA, but not greater, as required.

Hence, $R_{C2} = (V_{CC} - (V_{OL} + V_{BE4})) / I_{E2} = \quad (0 + 2.1 - 0.8) / 4.2 \times 10^{-3} = 309$ ohms

b) Typical circuit



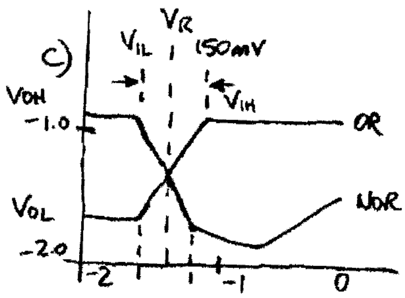
R_7, R_8, D_1 and D_2 are a potential divider to set V_{B5} to (–1.5 + 0.8).

D_1, D_2 enhance temperature stability

T_5 is an emitter follower ; R_6 should be chosen for convenient reference current, a few mA

$V_{B5} \sim V_{CC} - (V_{CC} - 2V_{diode} - V_E)\, R_7/(R_7 + R_8)$

c)



$V_{IH} = -1500 + 150/2$ mV $= -1.425$ V

$V_{IL} = -1500 - 150/2$ mV $= -1.575$ V

$N_{MH} = V_{OH} - V_{IH} = -0.9 + 1.425 = 0.525$ V

$N_{ML} = V_{IL} - V_{OL} = -1.575 + 2.1 = 0.525$ V

d) If the h_{FE} of T_4 is 50, then $I_{B4} = I_{E4}/51$. I_{B4} flows through R_{C2} and slightly reduces V_{B4} and hence V_{OH}, so slightly eroding the noise margin. If the gate does not drive any other gates:

$I_{E4} = (V_{OH} - V_{EE}) / R4 = \quad (-0.9 + 6) / 50 \times 10^3 \quad \sim \quad 102\ \mu A$

Hence $I_{B4} = 102/51 \quad = 2\ \mu A$

This extra current flowing through R_{C2} will increase the voltage drop in this component by $2 \times 10^{-6} \times 309 \sim 600\ \mu V$, hence reducing V_{OH} to –0.90006 V. This difference is quite negligible.

e) Each driven stage consumes a small amount of current from the output of the driving stage. The greater the current, the greater the fall in drive voltage, until the noise margin drops to an unacceptable level.

We are told that the noise margin may fall by 525/10 mV owing to loading effects. We do not actually know how much current each subsequent gate requires at its input, so we must estimate, and assume that these hypothetical gates will be of the same family with $I_E \sim 5$ mA and $h_{FE} \sim 50$, from which we can deduce that the required base current at each input is ~ 100 μA.

Owing to the emitter followers (T_4 and T_3), the increase in base current I_{B4} due to each additional driven input is about $100 \times 10^{-6} \times R2 / 51$. Taking the worst case, with $R_2 = 309$ ohms, Delta V_{OH} per additional input ~ 620 μV.

If a reduction of 52 mV is allowable, this corresponds to 52/0.62 ~ 80 driven gates.

In practice the net capacitance of the set of driven gate inputs plus the associated parasitic capacitances due to wiring, PC tracks, connectors, etc will limit the switching times achievable and the fanout will need to be restricted to a much lower value determined by these considerations.

Examiner's note: most candidates

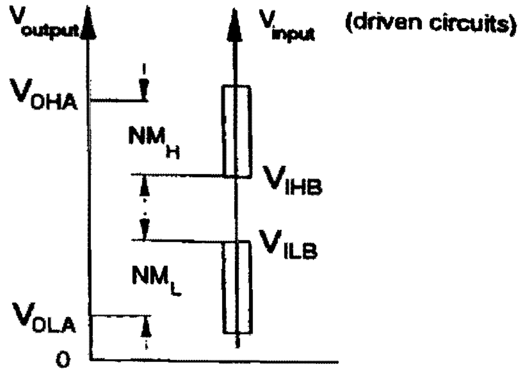
2 (a) **Main reasons for popularity of CMOS**:

- Gate inputs are effectively open-circuit, very high impedance, easy to drive
- Power supply current for static/low frequency apps is very low, ideal for battery-powered portable devices
- Very good noise immunity, ~0.4 V_{DD} for inverter both low and high states (not quite so high for multi-input gates)
- Fully restored logic levels V_{DD} and 0V
- Can operate over wide range of supply voltages
- Creates little electrical noise
- Easily integrated with linear circuitry for complex mixed-signal designs

Main disadvantages:

- Not as fast as GaAs or some forms of bipolar/BiCMOS
- Comparatively sensitive to static breakdown
- Liable to destructive latch-up as compound doped layers form thyristor-like structures

(b)



V_{OHA} and V_{OLA} represent respectively:

- the lowest voltage supplied by logic circuit A delivering logic '1', and
- the highest voltage supplied by an output delivering logic '0'

V_{IHB} and V_{ILB} represent respectively:

- the lowest input to B acceptable as logic '1'
- the highest input to B acceptable as logic '0'

If the output of A is connected to the input of B, the noise margins observed in the High and Low states are:

$$NM_H = V_{OHA} - V_{IHB} \qquad \text{and} \qquad NM_L = V_{ILB} - V_{OLA}$$

Both noise margins must be positive if the pair of circuits is to operate consistently. Their magnitude must be δ or greater if superimposed noise of voltage magnitude up to δ is to be rejected.

Note that the voltages V_{OH} and V_{OL} are liable to depend on the magnitude of the current flowing in the corresponding output devices – i.e. they depend on fan-out.

(c) The RAM cell consists of a pair of cross-coupled inverters (M2, M3 and M4, M5). This forms a bistable unit which can be in either of two stable states corresponding to a stored **logic 0** or **logic 1**.

If inverter (M2, M3) generates **logic 1**, that will cause inverter (M4, M5) to output **0**, which will cause (M2, M3) to output **1** … so maintaining the original values stable.

A similar argument holds if inverter (M2, M3) generates **logic 0**, giving **logic 1** at the output of (M4, M5).

Hence the cell may assume only two states, and if isolated from external influences will retain its state while power is applied.

MOSFETs M1 and M6 allow the inverter inputs to be accessed from outside for writing a value (**0** or **1**), and the outputs, for reading. Both are controlled by the ‘word’ line which would normally be controlled by decoding an address bus.

The state may be changed by driving the ‘bit’ and ‘$\underline{\text{bit}}$’ lines to new complementary values and operating switching transistors M1 and M6 by setting ‘word’ **high**. The new value is forced on to the inputs/outputs, of the two inverters, which assume the new state.

To read out data, both bit lines are preset to precisely the same potential, typically $V_{DD}/2$. They are fed to the inputs of a sensitive comparator, which will initially indicate the equivalence between the lines. The switching MOSFETS M1, M6 are then enabled. One inverter will drive its corresponding but line high, the other low, by a few mV. This is because the inverters are by design weak, and the bit lines are a substantial capacitive load.

The comparator then indicates 0 or 1 according to the sense of the perturbations introduced to the bit lines.

When the ‘word’ lines are deselected, M1 and M6 cease conducting and the inverters revert to their former logic values.

This only works if:

(a) the gates driving ‘bit’, ‘$\underline{\text{bit}}$’ are strong (low impedance sources)

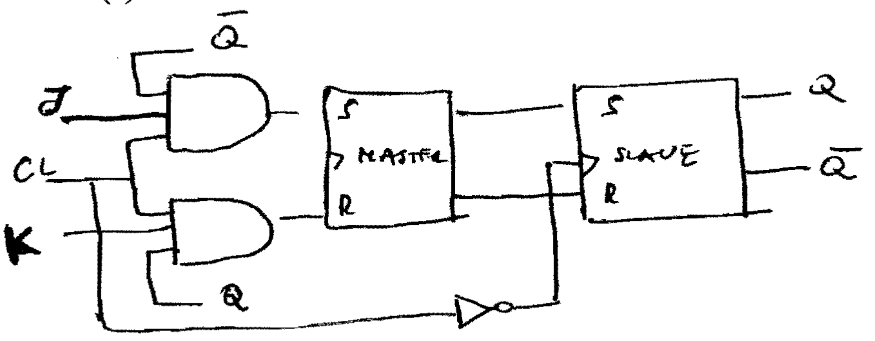
(b) M1 and M6 are sufficiently conductive when ‘word’ is selected

(c) the devices M2, M3 and M4, M5 are weak (high impedance), so can be overridden by ‘bit’, ‘$\underline{\text{bit}}$’ applied via M1, M6.

Hence M1, M6 have a relatively large W/L, while M2-M5 have much smaller W/L. This is also consistent with the requirement for the cell to be compact.

Examiner's note: this question ...

3. (a)



$$Q^+ = J\overline{Q} + \overline{K}Q,$$

Master sets up as clock rises – slave follows as clock falls.

[30%]

(b) By using the reverse engineering method from Fig 1 we can obtain:

$J_0 = Q_2Q_1 + \overline{Q_2}\overline{Q_1}$
$K_0 = Q_2\overline{Q_1} + \overline{Q_2}Q_1$
$J_1 = \overline{Q_2}Q_0$
$K_1 = Q_2Q_0$
$J_2 = Q_1\overline{Q_0}$
$K_2 = \overline{Q_1}\overline{Q_0}$

In a J-K bistable the next state Q^+ function of the present state Q is given by :
$Q^+ = J\overline{Q} + \overline{K}Q$
Thus we can calculate the next states Q_0^+, Q_1^+, Q_2^+

$Q_0^+ = Q_2Q_1 + \overline{Q_2}\overline{Q}1$
$Q_1^+ = \overline{Q_2}\overline{Q_1}Q_0 + \overline{Q_2}Q_1 + \overline{Q_0}Q_1$
$Q_2^+ = Q_1\overline{Q_0}\overline{Q_2} + Q_1Q_2 + Q_0Q_2$

The k-maps obtained from the expressions above will help to build the state table:

Q_0^+

$Q_2 Q_1$ \ Q_0	0	1
00	1	1
01	0	0
11	1	1
10	0	0

Q_1^+

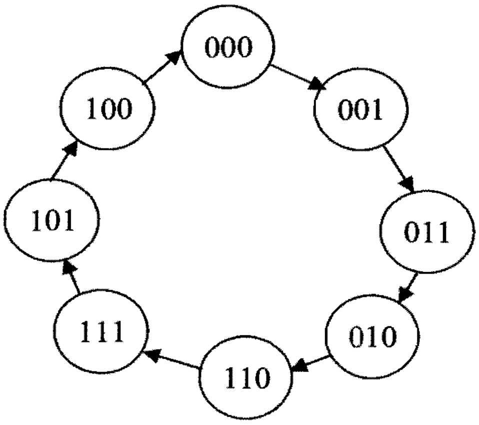
$Q_2 Q_1$ \ Q_0	0	1
00	0	1
01	1	1
11	1	0
10	0	0

Q_2^+

$Q_2 Q_1$ \ Q_0	0	1
00	0	0
01	1	0
11	1	1
10	0	1

From these K-maps we can build the state table and the state diagram. Since this is a counter (there are no inputs) it makes sense to write the states in the counting sequence:

Present state	Next state
000	001
001	011
011	010
010	110
110	111
111	101
101	100
100	000
000	

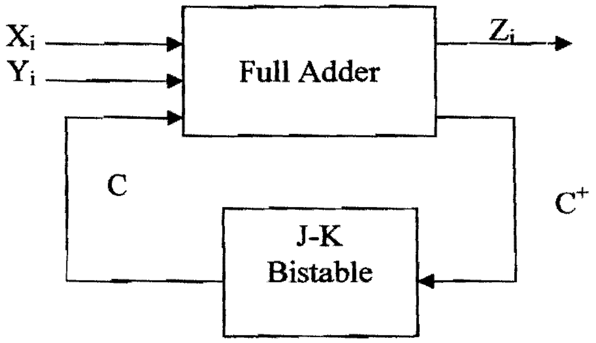


[50%]

One can see that from the one state to another only one bit changes at a time. This is therefore a 3-bit Gray code counter !

[25%]

4. (a) Output is only a function of the present state while next state (Q^+) is a function of both present state and primary inputs.



C = carry bit (present state)

C^+ = next carry bit (next state)

X_iY_i	C	Z_i	C^+	JK
00	0	0	0	0x
00	1	1	0	x1
01	0	1	0	0x
01	1	0	1	x0
10	0	1	0	0x
10	1	0	1	x0
11	0	0	1	1x
11	1	1	1	x0

J

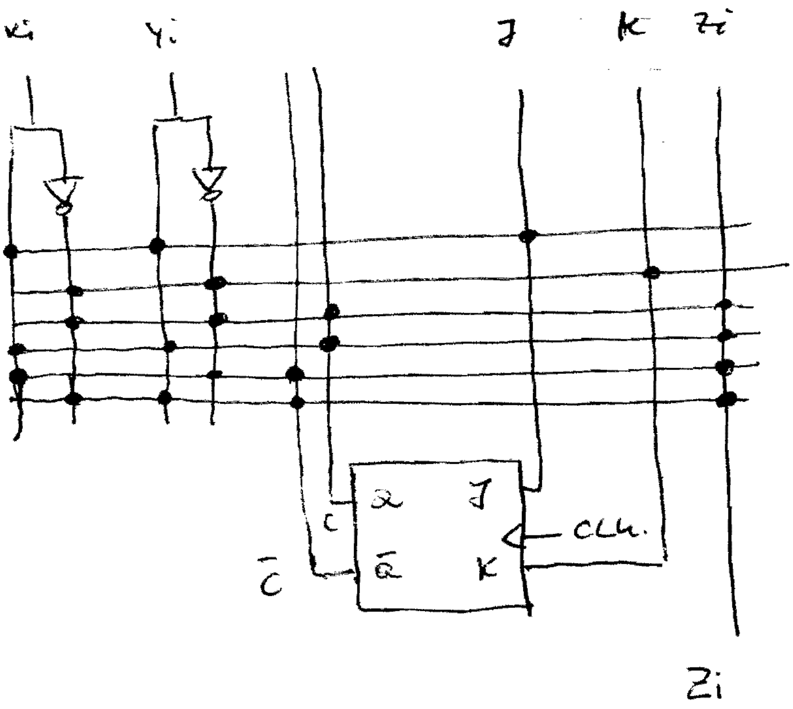
	00	01	11	10
0	0	0	1	0
1	×	×	×	×

$J = X_iY_i$

K

	00	01	11	10
0	×	×	×	×
1	1	0	0	0

$K = \overline{X_i}\,\overline{Y_i}$



Z_i

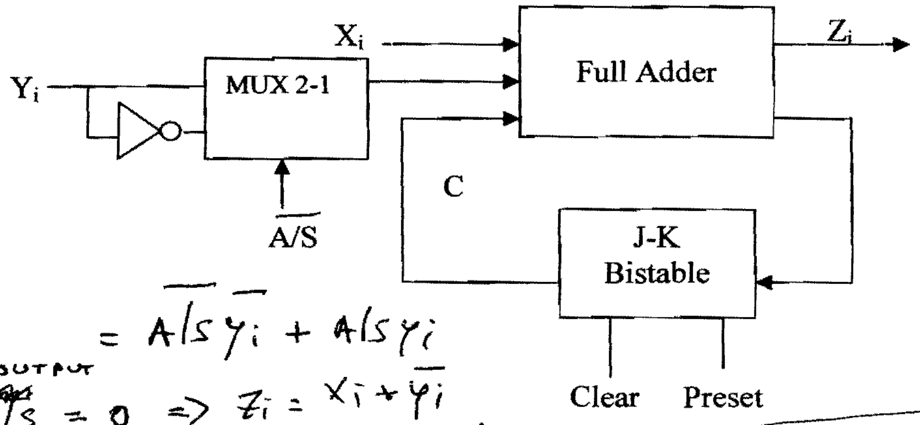
C \ X_iY_i	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$Z_i = X_i \oplus Y_i \oplus C$

$Z_i = \overline{X_i}\,\overline{Y_i}\,C + X_iY_iC + \overline{X_i}Y_i\overline{C} + X_i\overline{Y_i}\,\overline{C}$

[30%]

(b)



$\text{MUX}|_{\text{OUTPUT}} = \overline{A/S}\,\overline{Y_i} + A/S\,Y_i$

- when $\overline{A/S} = 0 \Rightarrow Z_i = X_i + \overline{Y_i}$
  if PRESET = 1 for the first bit only $Z_0 = X_0 + \overline{Y_0} + 1$ which means the operation is $Z = X - Y$
- when $A/S = 1 \Rightarrow Z_i = X_i + Y_i$
  $Z = X + Y$
- Replace in (a) for J, K, Z_i Y_i with $\overline{\text{MUX}}|_{\text{OUTPUT}}$

$$\Rightarrow J = X_i\,\overline{A/S}\,\overline{Y_i} + X_i\,A/S\,Y_i$$
$$k = \overline{X_i}\left(\overline{A/S}\,Y_i + A/S\,\overline{Y_i}\right)$$

$$\begin{aligned} Z_i = {} & \overline{X_i}\,\overline{(\overline{Y_i}\,\overline{A/S} + A/S\,Y_i)}\,C + \\ & X_i(\overline{A/S}\,\overline{Y_i} + A/S\,Y_i)\,C + \\ & \overline{X_i}(\overline{A/S}\,\overline{Y_i} + A/S\,Y_i)\,\overline{C} + \\ & X_i\,\overline{(\overline{A/S}\,\overline{Y_i} + A/S\,Y_i)}\,\overline{C} \end{aligned}$$

The implementation is similar to (a) with 3 inputs $X_i, Y_i, A/$ instead of the two X_i, Y_i.

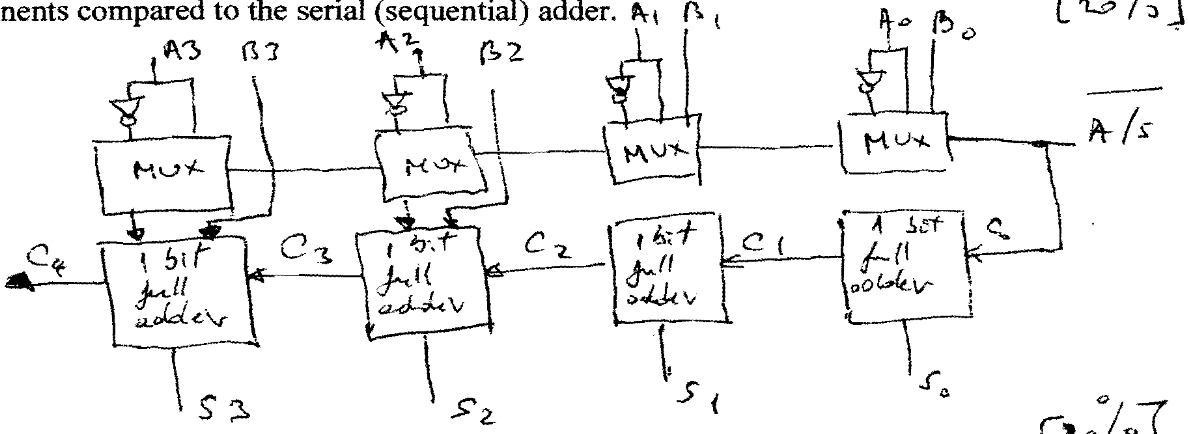
[30%]

(c) For 4 bit numbers the sequential adder will require four consecutive clock cycles to perform the 4-bit addition. The combinational ("parallel" or " ripple- carry") adder wills perform the addition in a simple step which equates to a single clock cycle.

- When the number of bits increases (e.g. 32 bits) the differences between the two designs become more prominent. The parallel adder will require a huge number of components compared to the serial (sequential) adder.

[20%]

(d)



[20%]