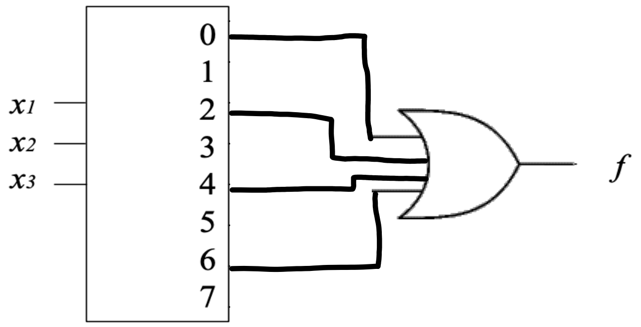
3B2 2023 CRIBS

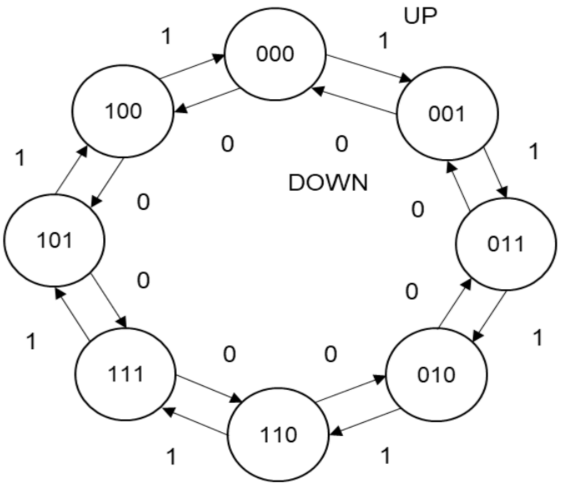
1 (a) Multiplexers and LUTs are suitable for single-output functions. LUT inputs are select lines of cascaded multiplexers, and can produce any function of n-inputs (addresses). ROMs are better for multiple-output functions but they become expensive and less efficient for very high number of variables. PLAs are good for multiple-output functions with lots of variables. They are expensive unless the number of variable is too high for a ROM. [20%]

(b) The function $f(x_1, x_2, x_3) = \sum(0, 2, 4, 6)$ can be implemented using a 3-to-8 binary decoder and an OR gate.



[30%]

(c) (i)



Present State			Next State					
			DOWN ($Y = 0$)			UP ($Y = 1$)		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	1
0	1	1	0	0	1	0	1	0
0	1	0	0	1	1	1	1	0
1	1	0	0	1	0	1	1	1
1	1	1	1	1	0	1	0	1
1	0	1	1	1	1	1	0	0
1	0	0	1	0	1	0	0	0

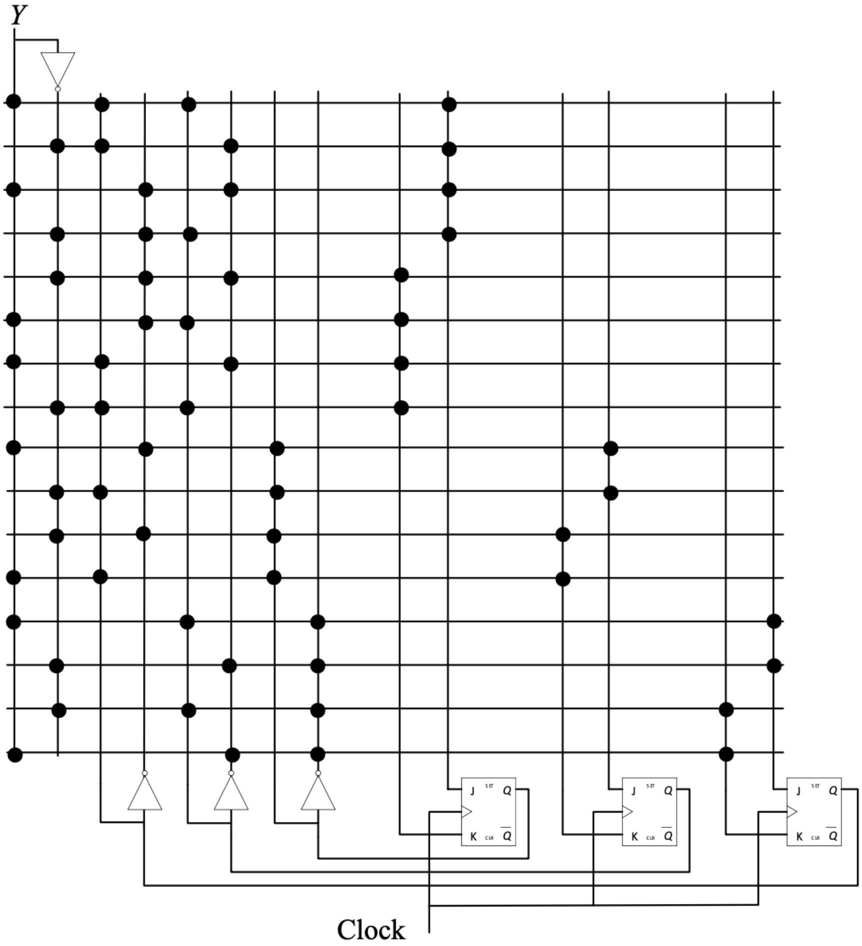
[20%]

(ii)

$J_0 = Q_2Q_1Y + Q_2\overline{Q_1}\,\overline{Y} + \overline{Q_2}\,\overline{Q_1}Y + \overline{Q_2}Q_1\overline{Y}$ $\qquad K_0 = \overline{Q_2}\,\overline{Q_1}\,\overline{Y} + \overline{Q_2}Q_1Y + Q_2\overline{Q_1}Y + Q_2Q_1\overline{Y}$

$J_1 = \overline{Q_2}Q_0Y + Q_2Q_0\overline{Y}$ $\qquad K_1 = \overline{Q_2}Q_0\overline{Y} + Q_2Q_0Y$

$J_2 = Q_1\overline{Q_0}\,\overline{Y} + \overline{Q_1}\,\overline{Q_0}\,\overline{Y}$ $\qquad K_2 = Q_1\overline{Q_0}\,\overline{Y} + \overline{Q_1}\,\overline{Q_0}Y$



[30%]

Assessor's Comments: This was a very popular question with 79 attempts out of a total of 82 students. The mean was 67.4 out of 100 with a standard deviation of 16.9. The part (a) was about the main principles and differences of main building blocks of digital circuit and FPGA such as multiplexers, LUTs, ROMs, PLAs. Most students clearly explained the advantages and disadvantages of these components. Part (b) was about assessing the basic understanding of a given VHDL code, which included a logic function. The students were mostly very successful reading the VHDL code. Part (c) focused on design and J-K bistable implementation of 3-bit up/down Gray code counter. Most of the students obtained the truth table and design correct. Majority of the students accurately determined the Boolean functions for J-K bistables. Overall, the students showed that they grasped all basic concepts regarding the FPGAs, VHDL, deriving logic functions and their implementations.

2 (a) In a Mealy network, the primary outputs, Z=f(x,Q), is a function of primary inputs, x, and present states, Q, while in a Moore network, Z=f(Q) is a function of the present states only. A main difference compared to the Mealy network is that when a set of inputs is applied to the Moore network, the resulting outputs do not appear until after the clock pulse causes the flip-flops to change state. [10%]

(b) It is a Moore configuration. Z=f(Q)

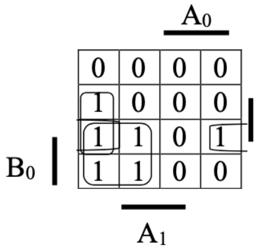
Z	
x	Q
0	1
1	1

The output, Z = x'Q + xQ = Q(x'+x) = Q, depends on Q only.

[20%]

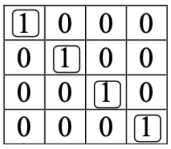
(c) (i)

$A_0A_1B_0B_1$	x y z
0 0 0 0	0 1 0
0 0 0 1	1 0 0
0 0 1 0	1 0 0
0 0 1 1	1 0 0
0 1 0 0	0 0 1
0 1 0 1	0 1 0
0 1 1 0	1 0 0
0 1 1 1	1 0 0
1 0 0 0	0 0 1
1 0 0 1	0 0 1
1 0 1 0	0 1 0
1 0 1 1	1 0 0
1 1 0 0	0 0 1
1 1 0 1	0 0 1
1 1 1 0	0 0 1
1 1 1 1	0 1 0



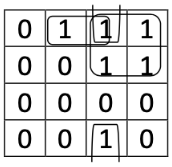
0	0	0	0
1	0	0	0
1	1	0	1
1	1	0	0

K-map for x



1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

K-map for y



0	1	1	1
0	0	1	1
0	0	0	0
0	0	1	0

K-map for z

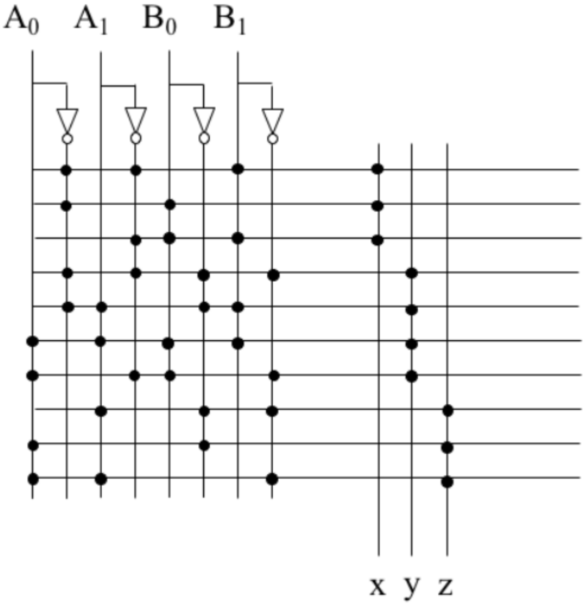
$$x=\overline{A_0}\,\overline{A_1}B_1+\overline{A_0}B_0+\overline{A_1}B_0B_1$$

$$y=\overline{A_0}\,\overline{A_1}\,\overline{B_0}\,\overline{B_1}+\overline{A_0}A_1\overline{B_0}B_1+A_0A_1B_0B_1+A_0\overline{A_1}B_0\overline{B_1}$$

$$z=A_1\overline{B_0}\,\overline{B_1}+A_0\overline{B_0}+A_0A_1\overline{B_1}$$

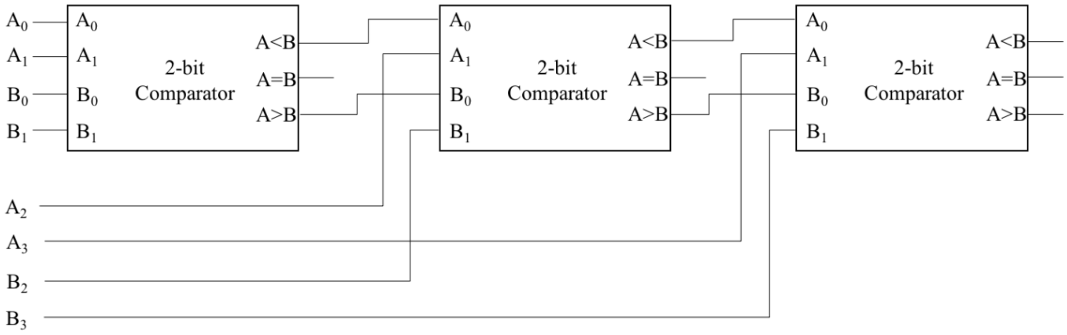
[30%]

(ii)



The size of a PLA implementation (10x3 above) is comparable to that of a needed ROM (16x3). A ROM might offer a more economical solution. [20%]

(d)



We first compare the lower two bits of A and B, then next and next bits.

Example: Compare A=1011 and B=1010

We first compare 11 (A_1A_0) and 10 (B_1B_0) and we get a 1 at (A>B). Hence, on the next comparator, we put a1 at B_0 and a0 at A_0 and A_2 and B_2 at A_1 B_1 pins. So, if A_2 is greater than B_2 then we get a 1 at (A>B) for 3 bits and if A_2 is less than B_2 we get a 1 at (A<B) for 3 bits and if A_2 is equal to B_2 then we compare A_0 and B_0. Similarly, we repeat to get the result. [20%]

Assessor's Comments: This question was about the fundamental types of sequential logic circuits and their implementations. More specifically, part (a) asked for Mealy and Moore logic and their differences and part (b) has given a circuit composed of a 2-to-4 decoder and a D bistable asking if it is Mealy and Moore logic with explanations. It was pleasing that many candidates could answer both parts correctly. The main concepts of 2-bit comparator logic were examined in part (c), which asked for its truth table, output functions and implementations using either a ROM or a PLA. Many students have successfully designed and implemented the circuit for 2-bit comparator. Part (d) focused on implementing a 4-bit comparator circuit using the given comparator in Figure 2. Few students tried to do it using other logic components although the question clearly asked for design using only 2-bit comparator circuits. A large group of candidates performed this implementation correctly. Overall, the question was the most popular one attempted by almost all the candidates, i.e., 80 out of 82, and, pleasingly answered very well with the mean of 64.4 out of 100.

3

(a) NMOS uses just n-type field-effect transistors in a pull-down network representing the logic function and acts against high-side resistors. CMOS in contrast has a n-type pull-down network (similar to NMOS) and a p-type pull-up network, which is logically inverse. In contrast to CMOS, NMOS has a nonnegligible static power dissipation. Ass the pull-down network furthermore acts against (linear) resistors, NMOS is typically slower, and the capacitance of the next stage may be discharged rather fast (through the transistors) but charged up only slowly (through the resistors) so that the behaviour is asymmetric. Thus, while CMOS is typically faster and lower power, NMOS can have a lower cost and only needs one type of transistors (smaller n-type transistors).

The two fundamental ingredients for a CMOS circuit are p-type and n-type field-effect transistors. The mobility of electrons is approximately twice as high as that of holes in silicon so that p-type transistors need twice the channel width for comparable current and therefore matched pull-up and pull-down performance (statically and dynamically). For stable levels and low noise influence, higher voltages would be preferred. However, high voltages lead to higher loss (approx. ~CV^2 after all).

(b)

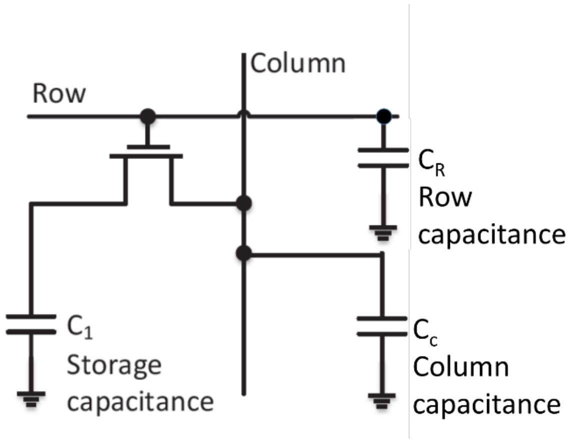
A key design choice refers to the pull-up resistors. The smaller they are, the faster the circuit but the larger the static power loss. The pull-down network are simply n-type field-effect transistors of any kind. The pull-up network is formed by resistors. The latter can be implemented in various ways. Small and medium resistors can be generated as meandering paths through silicon. Alternatively, transistors (preferably n-type in NMOS) with the gate connected to a fixed potential, e.g., drain.

(c)

TTL uses BE junctions as inputs followed by a gain stage and an output driver. Input transistor even operated in inverse mode for low output. Gain stage easily saturates for low output. Input transistors saturate for high output. For any output state, at least one output transistor tends to saturate. High levels are substantially below the operating voltage as the voltage drops of the output current across a resistor, the pull-up output transistor, and a diode have to be considered.

ECL avoids saturation of transistors, which limits the speed TTL gates. For achieving that, ECL uses differential pairs of transistors coupled at their emitters, and the sum of the emitter current is limited intentionally through biasing one of the transistors with a reference voltage V_R. ECL is therefore one of the fastest logic technologies. Every output is typically provided with its inverse.

(d)



Whereas the row capacitance is (dis-)charged by a driver, the column capacitance can absorb part of the storage capacitor's charge, i.e., data. If it were as large as the storage capacitance, the read-out level could fall to half of the initial level just because of this effect.

(e)

(i)

$t_{LH} \sim C/(k\ V) = 1.7 \cdot 10^{-12+5}$ s = 0.17 µs >> 1/600 µs

=> k at least by a factor of 0.17 · 600 ~100 better, i.e., k > 10^{-3} A/V²

Alternatively, smaller capacitance

(ii)

$P_W = 600 \cdot 10^6$ Hz × 2 × 64 × ½ × ½ C V² = 53 mW

(iii)

Regular refresh => voltage practically constant. So most of the numbers not needed.

$P_D = 10^9$ × ½ × V²/R = 0.72 W

(3.3 V/1.2 V)² = 7.65 times the loss

Assessor's Comments: Question 3 was attempted by 57 students out of 82 and overall answered well. The focus of this question was knowledge and understanding with a relatively simple but again understanding-driven calculation at the end. Most students had no problems with parts (a) and (b), though a few candidates either mixed up technologies. Similarly, most students had few issues with part (c), although a number of students could not explain or clearly did not know why TTL and ECL have different speed. In part (d), some students solved the question exemplarily, demonstrating that the topic was indeed discussed in the course, but a number of students had either not understood the problem of parasitic capacitances fully or could not recall, which one is the most problematic one as it absorbs the data signal. Only few students were not able to sketch a memory cell at all. In part (e), the difficulty was to understand that one part was aiming at the dynamic loss (charging up and down the capacitance of the memory cells), the other one the leakage. A number of students overlooked that difference and, for example, considered the leakage resistance to be the resistance through which the memory cell is charged and estimated an exceptionally high time constant. Once this difference was found by a student, as still many did, the remaining math and physics was rather simple.

4

(a)

Emitter-coupled OR and NOR gate.

(b)

Swing of 1.2 V symmetrical around V_R.

=> $V_{OH} = V_R + 1.2\ \text{V}/2 = –0.9\ \text{V}$

=> $V_{OL} = V_R – 1.2\ \text{V}/2 = –2.1\ \text{V}$

For max. speed use max emitter current I_E

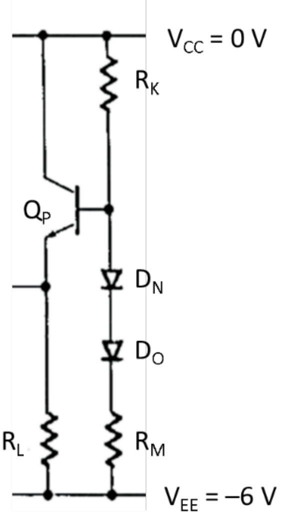
$V_{E1} = V_{OH} – V_{BE1} – V_{EE}$

$R_E = V_{E1}/I_{E1} = 1/I_{E1}\ (V_{OH} – V_{BE1} – V_{EE}) = 1/4\ \text{mA}\ (–0.9\ \text{V} – 0.65\ \text{V} + 6\ \text{V}) = 1112.5\ \text{k}\Omega$ $= 1.11\ \text{k}\Omega$

Choose R_1 so that NOR output at $V_{OL} = –2.1$ V when T_3 on and $I_{EI} = 4$ mA.

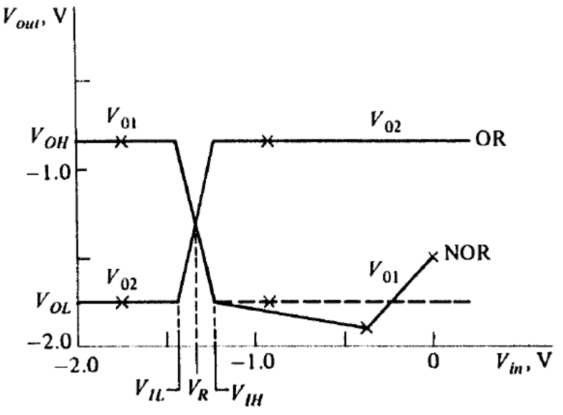
$R_1 = –(V_{OL} + V_{BE3})/I_{E1} = (2.1\ \text{V} – 0.65\ \text{V})/4\ \text{mA} = 363\ \Omega = R_2$

(c)



R_K and R_M form the potential for the base of Q_P, which Q_P "copies" to its emitter by one V_{BE} shifted. Diodes, such as D_N and D_O, introduce one or several pn junctions similar to the base-emitter one in Q_P to compensate some of the temperature dependence of the output.

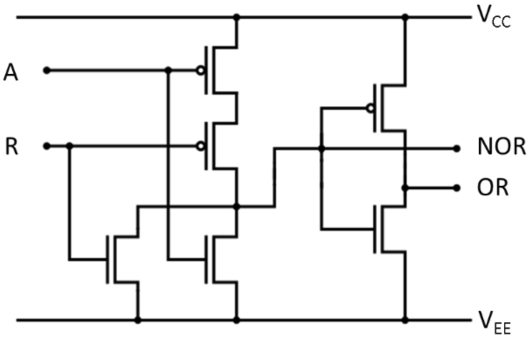
(d)



$V_{IH} = –1.5 \text{ V} + 150 \text{ mV}/2 = –1.425 \text{ V}$
$V_{IL} = –1.5 \text{ V} – 150 \text{ mV}/2 = –1.575 \text{ V}$

$N_{MH} = V_{OH} – V_{IH} = –0.9 \text{ V} + 1.425 \text{ V} = 0.525 \text{ V}$
$N_{ML} = V_{IL} – V_{OL} = –1.575 \text{ V} + 2.1 \text{ V} = 0.525 \text{ V}$

(e)



NOR gate with subsequent inverter as the most straightforward solution.

Assessor's Comments: Question 4 had a stronger focus on calculations for students who prefer this style. Most students also performed well here but made calculation mistakes early on so that they practically got lost on the way. The first question still started with knowledge, identifying the most likely most characteristic logic family based on the schematic in the figure. The majority of students had no problems with that. Part (b) required finding the component values for the resistors. Most students understood that they have to derive the logic levels first based on the given voltage swing. Unfortunately, a rather large portion of students did not divide the swing by two and ended up with rather large levels. Part (c) involved some knowledge of how to generate the reference voltage needed for ECL. In principle, the correct solution would use a voltage divider, a transistor for stabilising the level, and diodes to "copy" the pn junction and therefore its temperature behaviour. However, also a voltage divider (with diodes) alone was treated as correct. Part (d) was solved without any flaws by some students, while many others used wrong values from previous part (which was again treated with clemency as long it could be identified where the issue might have come from). Part (e) was overall answered well with mostly only confusion if OR/NOR or inverter gates would be the most appropriate equivalent (which both were as long as they were in CMOS technology). Few students mixed up the logic and, for instance, sketched NMOS logic.