

## Discussion on CS1104 Exam Questions 2 and 3 (AY2004/5 Semester 2)

Q2. a) Convert the quinary (base-5) value  $2101.4_5$  to its ternary (base-3) equivalent, correct to 4 ternary places.

Then, convert the value  $2101.4_5 \div 3^4$  to ternary base, correct to 3 ternary places.

### Discussion:

Convert  $2101.4_5$  to base 10 first, then, convert it to base 3.

$$2101.4_5 = 276.8_{10} = 101020.21012_3 = \mathbf{101020.2102_3}$$

(Why is this preferred over  $101020.2101_3$ ?)

There is no need to compute  $2101.4_5 \div 3^4$  the hard way; all you need is to move the ternary point of the answer above by 4 places to the left (why?):

$$2101.4_5 \div 3^4 = 10.10202102_3 = \mathbf{10.102_3}$$

### Common mistakes:

Here are some common mistakes spotted in my grading (errors in red):

- $2101.4_5 = (2 \times 5^3) + (1 \times 5^2) + (0 \times 5^1) + (1 \times 5^0) + (0.4 \times 5^{-1})$
- $2101.4_5 = (2 \times 5^4) + (1 \times 5^3) + (0 \times 5^2) + (1 \times 5^1) + (4 \times 5^{-1})$
- $2101.4_5 = (2 \times 5^3) + (1 \times 5^2) + (0 \times 5^1) + (1 \times 5^0) + (4 \times 5^{-1})$   
 $= 250 + 25 + 0 + 5 + 0.8$

b) A certain floating-point representation has 1-bit sign, 5-bit normalised mantissa, followed by 4-bit two's complement exponent. Given the following representations of two values  $A$  and  $B$ :

$$A = 0, 10100, 0010$$

$$B = 1, 10000, 1111$$

Compute the sum of  $A$  and  $B$ . Write out this sum in decimal form, and then in the above floating-point representation.

### Discussion:

$$A = 0.101 \times 2^2 = 0.625_{10} \times 4_{10} = 2.5_{10}$$

$$B = -0.1 \times 2^{-1} = -0.5_{10} \times 0.5_{10} = -0.25_{10}$$

$$A + B = \mathbf{2.25_{10}} = 10.01_2 = 0.1001 \times 2^2 = \mathbf{0, 10010, 0010}.$$

### Common mistakes:

A number of students do not have the right understanding of floating-point representation. Common mistakes include:

- Taking the exponent of  $A$  to be  $-14$ . Note that  $0010_{2s} = 0010_2 = 2_{10}$  so it should be 2, not  $-14$ .
- Taking the base of the exponents to be 10, or  
 $A = 0.101 \times 10^2 = 0.625_{10} \times 100_{10} = 62.5_{10}$   
 $B = -0.1 \times 10^{-1} = -0.5_{10} \times 0.1_{10} = -0.05_{10}$

- c) Given the following function table of a certain logic device that takes in 6-bit input  $ABCDEF$  and produces 3-bit output  $PQR$ , write out the minimal SOP expressions for  $P$ ,  $Q$  and  $R$ .

$A$	$B$	$C$	$D$	$E$	$F$	$P$	$Q$	$R$
X	X	X	X	X	0	0	0	0
X	X	X	X	0	1	0	0	1
X	X	X	0	1	1	0	1	0
X	X	0	1	1	1	0	1	1
0	0	1	1	1	1	1	0	0
1	0	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1

*Discussion:*

The way to derive the expression for an output from a compact truth table was discussed in Tutorial #5 Question 4.

$P = C.D.E.F$  (directly from observation from the truth table; or you may start with  $A'.B'.C.D.E.F + A'.B.C.D.E.F + A.B'.C.D.E.F + A.B.C.D.E.F$  and simplify from there.)

$$\begin{aligned} Q &= D'.E.F + C'.D.E.F + B.C.D.E.F \\ &= D'.E.F + C'.E.F + B.C.E.F \text{ (absorption theorem)} \\ &= D'.E.F + C'.E.F + B.E.F \text{ (absorption theorem)} \end{aligned}$$

$$\begin{aligned} R &= E'.F + C'.D.E.F + A.C.D.E.F \\ &= E'.F + C'.D.F + A.C.D.F \text{ (absorption theorem)} \\ &= E'.F + C'.D.F + A.D.F \text{ (absorption theorem)} \end{aligned}$$

*Common mistakes:*

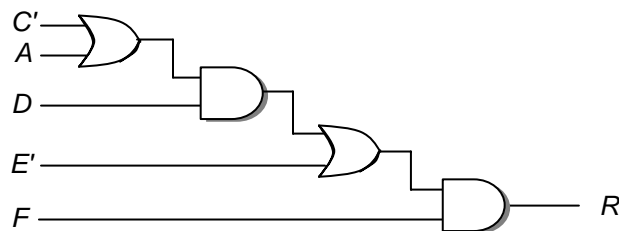
Many students did not apply the absorption theorem thoroughly; some did not even use the absorption theorem at all!

- d) You are given only 2-input AND gates and 2-input OR gates. Implement  $R$  in part (c) above using the fewest number of these gates, assuming that primed literals are available. Draw the logic circuit.

*Discussion:*

Just rewrite the expression for  $R$  obtained above into:

$$R = E'.F + C'.D.F + A.D.F = (E' + (C'+A).D).F$$



Q3.

- a) The logic circuit below shows a code converter that takes in a 4-bit code  $ABCD$  and converts it into a 4-bit code  $EFGH$ . It consists of two 4-bit magnitude comparators, a 4-bit parallel adder, and a quadruple 2:1 multiplexer devices. The output  $V$  is 1 if  $ABCD$  is a valid code, or 0 otherwise.

Analyse the circuit and complete the given truth table. From the truth table, deduce what kind of code converter it is.

*Discussion:*

Analyse the given circuit and complete the following truth table. You should not have problem if you understand the working of all the given devices.

$A$	$B$	$C$	$D$	$P$	$V$	$E$	$F$	$G$	$H$
0	0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	0	1
0	0	1	0	0	1	0	0	1	0
0	0	1	1	0	1	0	0	1	1
0	1	0	0	0	1	0	1	0	0
0	1	0	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	0	0
0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	1	1	1	1	0
1	0	0	1	1	1	1	1	1	1
1	0	1	0	1	0	0	0	0	0
1	0	1	1	1	0	0	0	0	1
1	1	0	0	1	0	0	0	1	0
1	1	0	1	1	0	0	0	1	1
1	1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	1	0	1

If you get the table above filled, you get most of the mark. Many students managed to do this correctly.

From the truth table, we can deduce that the output  $EFGH$  represents 2421 code. So the circuit is a **BCD to 2421 code converter**.

Common mistakes:

Some students filled in don't care values for the last 6 rows of  $E$ ,  $F$ ,  $G$ , and  $H$ , which is incorrect (but you won't suffer great loss) – why?

Q3. (continued)

- b) Implement a full adder with inputs  $X$ ,  $Y$ , and  $Z$  and outputs  $C$  (carry) and  $S$  (sum) using two 4:1 multiplexers with no additional gate, assuming that primed literals are available.

*Discussion:*

I believe if the question is changed to:

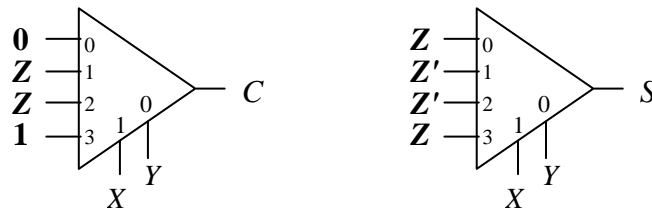
Given the truth table below, where  $X$ ,  $Y$  and  $Z$  are the inputs and  $C$  and  $S$  the outputs, implement  $C$  and  $S$  using two 4:1 multiplexers with no additional gate, assuming that primed literals are available.

Then you should be able to do it?

$X$	$Y$	$Z$	$C$	$S$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

But this is exactly the same question as the one given. The truth table above is that of a full adder, and it is given in your book and notes!

So, one possible answer is shown below. As you know, there are alternative answers since you can choose any two variables for the selector inputs.



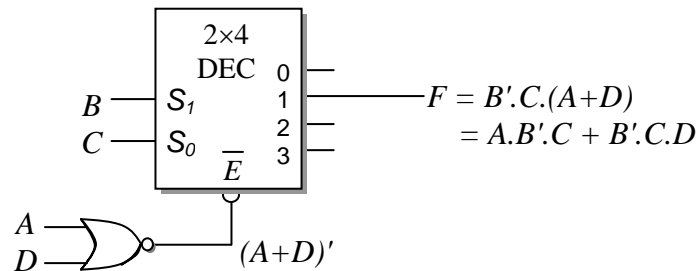
- c) Implement the function  $F(A,B,C,D) = \Sigma m(3, 10, 11)$  using a single  $2 \times 4$  decoder with zero-enable and at most one of these logic gates: inverter, AND, OR, NAND, NOR, XOR, XNOR. Primed literals are not available. [5 marks]

*Discussion:*

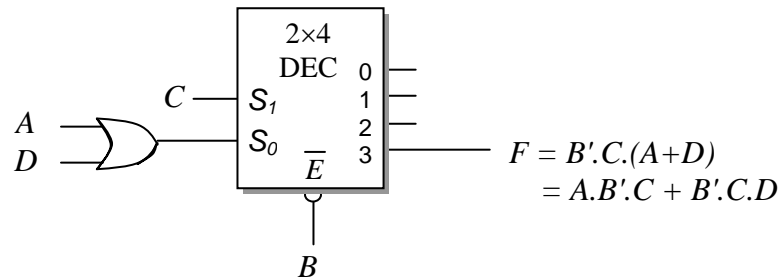
This could be the only part that requires a bit of extra thinking. The trick is to find a simplified expression for the given function, as simplified expression is always easier to analyze and work with.

$$\begin{aligned} F(A, B, C, D) &= A'.B'.C.D + A.B'.C.D' + A.B'.C.D \\ &= A.B'.C + B'.C.D \end{aligned}$$

From the above simplified SOP expression, we can derive the implementation.



or



There are, of course, other possible answers.

**Common mistakes:**

For students who attempted this question (many didn't), one common mistake is that they did not provide the enable input, or forgot that it is a zero-enable input.

*Note that partial credits can be given for an incomplete or incorrect answer, as long as the answer contains some deserving point. Hence, it is often advisable to show some working or intermediate steps, unless you are very sure of the correctness of your answer.*