**40001 Intro to Computer Systems 2021 Exam Sample Solution**

*Disclaimer: This is by no means an official answer key, please correct any mistakes you find and keep in mind there could be several possible solutions*

**Section A**

1 a. (A’ • (B + A)) + (B’ • (A + A’))

**≡** (A’ • (B + A)) + (B’ • 1) **[A + A’ = 1]**

**≡** (A’ • (B + A)) + B’ **[A • 1 = A]**

**≡** (A’ • B + A’ • A) + B’ **[Distributivity of •]**

**≡** (A’ • B + 0) + B’ **[A • A’ = 0]**

**≡** (A’ • B) + B’ **[A + 0 = A]**

**≡** (B’ + A’) • (B + B’) **[Distributivity of** +**]**

**≡** (B’ + A’) • 1 **[A + A’ = 1]**

**≡** (B’ + A’) **[A • 1 = A]**

**≡** (A • B)’ **[De Morgan’s Law]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **C0** | **C1** | **O** | **Minterm** |
| 0 | 0 | 0 | 0 |  |
| 0 | 0 | 1 | 0 |  |
| 0 | 1 | 0 | 1 | A’ • C0 • C1’ |
| 0 | 1 | 1 | 0 |  |
| 1 | 0 | 0 | 1 | A • C0’ • C1’ |
| 1 | 0 | 1 | 0 |  |
| 1 | 1 | 0 | 0 |  |
| 1 | 1 | 1 | 0 |  |

b. i)

**Canonical Minterm Form:**

(A’ • C0 • C1’) + (A • C0’ • C1’)

ii) 3 NOT gates, 4 AND gates and 1 OR gate

A picture containing diagram

Description automatically generated iii) D is a 4-to-1 multiplexer:

Diagram

Description automatically generated c

d. i) 0xCAFE0000 = 1100 1010 1111 1110 0000 0000 0000 0000

**Sign Bit**: 1 = Negative

**Exponent**: 1001 0101 = 149 – 127 = 22

**Mantissa**: 111 1110 0000 0000 0000 0000 = 0.984375

= 1.984375 (Restore hidden bit)

= -1.984375 x 222

ii) = -8.323072 x 106

**Section B**

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction** | **Cycle** | **Transfers** | **Path** |
| ADD Rdest, Rsrc | E1 | A 🡨 Rsrc |  |
| E2 | B 🡨 Rdest |  |
| E3 | Rdest 🡨 ALUres; C 🡨 ALUcout | ALU=A+B; Cin=0 |
| COMPARE Rdest, Rsrc | E1 | A 🡨 Rsrc |  |
| E2 | B 🡨 Rdest |  |
| E3 | C 🡨 ALUcout | ALU=A-B; Cin=0 |

2 a.

b. SUBTRACT, AND, OR and XOR are all two-register instructions that are very similar to ADD, just with different ALU settings

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction** | **Cycle** | **Transfers** | **Path** |
| SKIP | E1 | PC 🡨 PC + 1 |  |

c.

The COMPARE instruction is used by SKIP, as the carry register is updated after COMPARE is executed which is then used as an input to the controller

The NOP instruction requires 0 cycles and has no register transfers. It could be usefully applied for timing purposes, such as producing a delay for an idle processor

Graphical user interface

Description automatically generated d.

Text

Description automatically generated

e.

1/512 = 1/8 \* 1/8 \* 1/8, with the resulting circuit being asynchronous