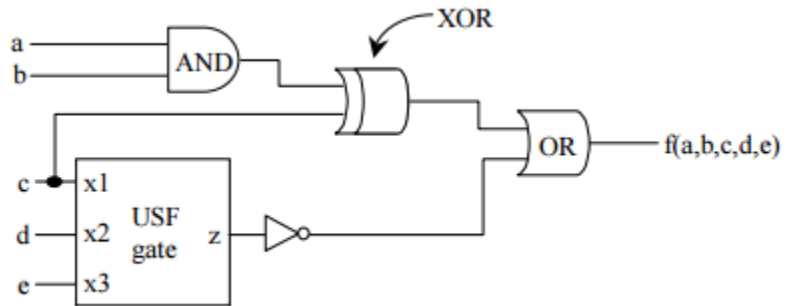


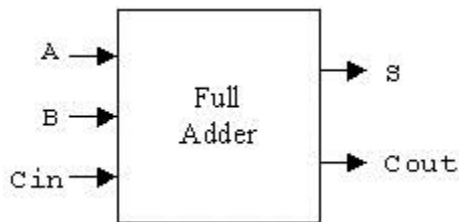
Review Questions:

1) You are given the below circuit. The gate labeled "USF gate" has the truth table also given below. Your job is to analyze the given circuit and come-up with a minimized two-level circuit for the same function that uses AND and OR gates. Draw the minimized two-level circuit.

x1	x2	x3	z
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



2) Derive the truth table for a 1-bit full adder



3) Consider a byte-addressable computer with 24-bit addresses, a 64KB data cache and a line size of 32 bytes. Show the format of a 24-bit address for:

- a) direct mapped
- b) associative
- c) 4-way set associative

4) Into what set does address 0x1234 map for 3 c)?

5) Give an example of subtracting an 8-bit number from another 8-bit number such that the result of the OF in the flags register is 0 and the result of the CF in the flags register is 1. Explain in detail what happened to cause this result.

6) Assume that the MIPS64 is a fully pipelined processor with 5 stages as shown in Figure #2. Further, the MIPS64 uses a predict taken strategy. For the given program below, answer the following questions:

```
.text
daddi r3,r0,0
daddi r4,r0,0
daddi r5,r0,5
L2:  dadd r4,r4,r3
     daddi r3,r3,1
     slt r1,r3,r5
     bnez r1,L2
     lwu r21,CONTROL(r0)
```

a) Assuming no forwarding, are there any data hazards in the above program? If so, identify one data hazard, the type of data hazard, and what causes the data hazard.

b) Assuming maximum forwarding, are there any data hazards in the above program? If so, identify one data hazard, the type of data hazard, and what causes the data hazard.

c) Are there any structural hazards? Why or why not?

Anything else you want to discuss?

Figure #2

MIPS64 Pipeline Stages

1. **IF (Instruction Fetch)**

MAR \leftarrow PC
IR \leftarrow M[MAR]

2. **ID/RF (Instruction Decode/Register Fetch)**

A \leftarrow Rs1
B \leftarrow Rs2
PC \leftarrow PC + 4
Note: Decoding can be done in parallel with reading registers

3. **EX (Execution)**

Memory Reference:
MAR \leftarrow A + immediate
MDR \leftarrow Data Read
ALU:
C \leftarrow A op B
C \leftarrow A op immediate
Branch
C \leftarrow new address
condition \leftarrow A op "0"

4. **MEM (Memory reference)**

Memory
MDR \leftarrow M[MAR] or
M[MAR] \leftarrow MDR
Branch:
if (condition) PC \leftarrow C

5. **WB (Write Back)**

Memory:

Rd <- MDR (through ALU)

ALU:

Rd <- C