Question 1 (20 points): For each of the following statements you have three options: (i) leave it blank; (ii) Mark it with **T** to indicate that the statement is true; (iii) Mark it with `F` to indicate that the statement is false. **You lose 2 points for each incorrect answer.** You win 4 points for each correct answer. You don't win or lose any points if you leave the statement blank. Regardless of how many statements you mark wrong, your score in this question cannot be below zero.

1. ( ) In an embedded processor whose design was derived from the MIPS architecture, the address field for a branch instruction is 8 bits. The execution procedure for a branch is the same as in MIPS:

```
bne  $s3, $s4, 8:      PC <-- PC + 4
                       if($1 != $2) then PC <-- PC + (address << 2)
```

   The largest distance that a branch can jump backward in this architecture is 128 instructions.

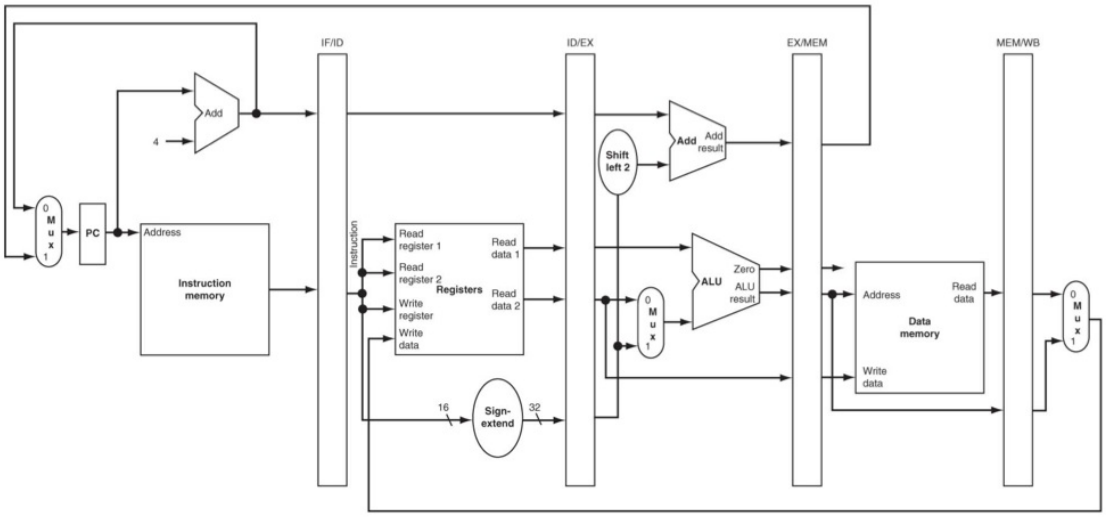
2. ( ) Assume two cache designs C_A and C_B have the same block size. C_A is a 16 KB 2-way set associative cache and C_B is an 8 KB direct-mapped cache. More bits are used for index in C_A than in C_B.

3. ( ) The following code correctly execute an atomic swap between the value stored in the address specified by `$s1` and the content of register `$s4`:

```
try:          add   $t0, $zero, $s4
              ll    $t1, 0($s1)
              sc    $t0, 0($s1)
              beq   $t0, $zero, try
              add   $s4, $zero, $t1
```

4. ( ) To ensure the correct operation of the system, an exception handler must save the value that it uses into a special frame stored in the stack of the program that got interrupted by the exception.

5. ( ) The figure below depicts the architecture of a 5-stage pipelined implementation of the data path for the MIPS architecture. This drawing is incorrect because the instruction fetched at time T_i will write to the register specified by the instruction fetched at time T_{i+3}.


IF/ID
ID/EX
EX/MEM
MEM/WB
Add
4
M
u
x
PC
Address
Instruction memory
Instruction
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
Shift left 2
Add
Add result
ALU
Zero
ALU result
Address
Read data
Data memory
Write data
16
Sign-extend
32