6 Introduction to Computer Architecture (swm11)

(*a*) For a microprocessor pipeline, what are control hazards and how can they be mitigated? [4 marks]

(*b*) Consider the 32-bit RISC-V assembler code below for the three functions `foo1`, `foo2`, and `foo3`. The RISC-V reference card has been provided if you need to look up an instruction.

```
foo1:
  mv   t0, zero
  beq  a0, zero, foo1exit
foo1loop:
  andi t1, a0, 1
  add  t0, t0, t1
  srli a0, a0, 1
  bne  a0, zero, foo1loop
foo1exit:
  mv   a0, t0
  ret

foo2:
  mv   s3, ra
  mv   s0, a0
  andi a0, s0, 0xff
  call foo1
  mv   s1, a0

  srli s0, s0, 8
  andi a0, s0, 0xff
  call foo1
  add  s1, s1, a0

  srli s0, s0, 8
  andi a0, s0, 0xff
  call foo1
  add  s1, s1, a0

  srli s0, s0, 8
  andi a0, s0, 0xff
  call foo1
  add  a0, s1, a0
  mv   ra, s3
  ret
```

```
foo3:
  li   t0, 0x55555555
  and  t1, a0, t0
  srli t2, a0, 1
  and  t2, t2, t0
  add  a0, t1, t2
  li   t0, 0x33333333
  and  t1, a0, t0
  srli t2, a0, 2
  and  t2, t2, t0
  add  a0, t1, t2
  li   t0, 0x0f0f0f0f
  and  t1, a0, t0
  srli t2, a0, 4
  and  t2, t2, t0
  add  a0, t1, t2
  li   t0, 0x00ff00ff
  and  t1, a0, t0
  srli t2, a0, 8
  and  t2, t2, t0
  add  a0, t1, t2
  andi t1, a0, 0xff
  srli t2, a0, 16
  add  a0, t2, t1
  ret
```

(*i*) If each function is called with the argument in register `a0=0xff000000`, what answer does it return? Explain your answers. [6 marks]

(*ii*) If this code was executed on a simple five-stage scalar pipeline (instruction-fetch, decode/register-fetch, execute, memory-access, write-back) with no branch prediction and branches taken in the execute state, how many clock cycles will each function take when called with `a0=0x0f` ? State your assumptions. [6 marks]

(*c*) A new custom instruction is proposed to speed up computation of `foo1`. Write an efficient combinational function for `foo1` in SystemVerilog. [4 marks]