RV32I Control Timing Diagrams

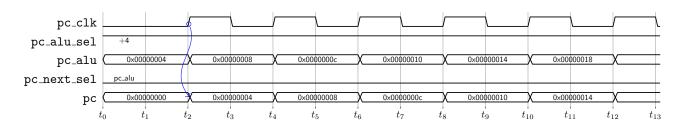
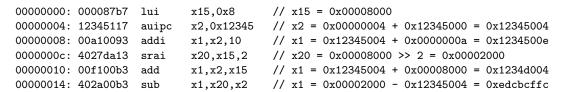


Figure 1: Program counter behavior.



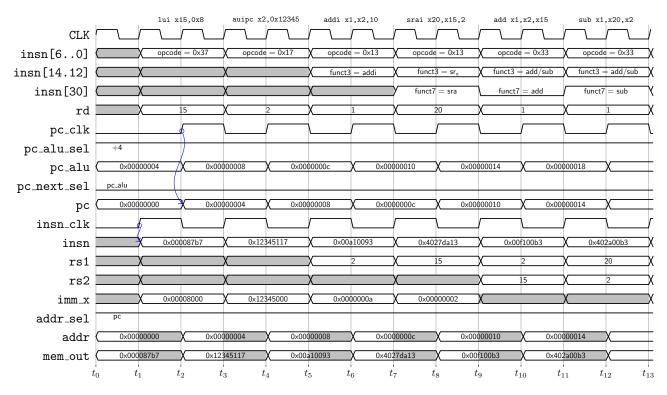


Figure 2: Fetch and decode cycles of U-type, R-type, and some I-type instructions.

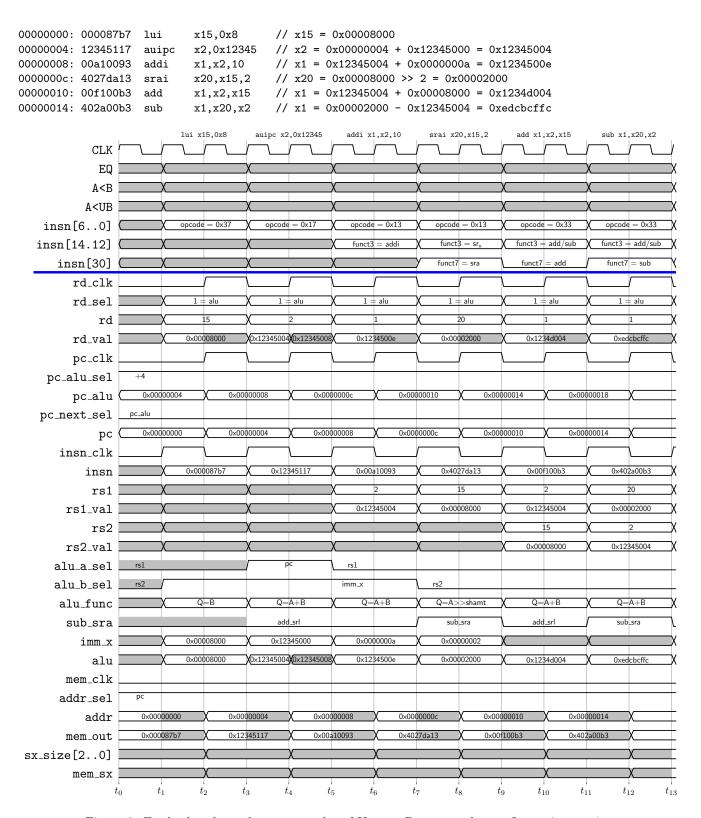


Figure 3: Fetch, decode, and execute cycles of U-type, R-type, and some I-type instructions.

```
// m32(0x00008000 + 0x00000008) = 0x12345004
00000018: 0027a423
                                 x2,8(x15)
                                 x16,8(x15)
0000001c: 0087a803
                                                 // x16 = sx(m32(0x00008000 + 0x00000008)) = 0x12345004
                                                 // x1 = 0x00000024, pc = 0x00000020 + 0x00000008 = 0x00000028
00000020: 008000ef
                                 x1,0x28
00000028: 03000267
                                 x4,48(x0)
                                                 // x4 = 0x0000002c, pc = (0x00000030 + 0x00000000) & 0xfffffffe = 0x00000030
                       jalr
                                 x0,x15,0x50 // pc += (0x000000000 == 0x000008000 ? 0x00000020 : 4) = 0x00000034
00000030: 02f00063
                       beq
                                 x2,x16,0x54 // pc += (0x12345004 == 0x12345004 ? 0x00000020 : 4) = 0x00000054
00000034: 03010063
                       beq
                                                                                                              beq x2,x16,0x54
                                                               jal x1,0x28
                                                                                              beq x0,x15,0x50
                              sw x2,8(x15)
                                              lw x16,8(x15)
                                                                              jalr x4,48(x0)
              CLK
               EQ
              A<B
            A<UB
    insn[6..0]
                               opcode = 0x23
                                               opcode = 0x03
                                                               opcode = 0x6f
                                                                                opcode = 0x67
                                                                                                opcode = 0x63
                                                                                                                opcode = 0x63
   insn[14.12]
       insn[30] (
          rd_clk
          rd_sel
               rd
          rd_val
          pc_clk
     pc_alu_sel
          pc_alu
                       0x0000001c
                                                        0×00000024
                                                                                         0x00000034
                                                                                                                 0×00000054
    pc_next_sel
                                                                                                         0×00000034
                                                                                                                    X0×00000054
               рс
                       0x00000018
                                                        0×000000020
        insn_clk
            insn
              rs1
                                                0×000080000
                                                                                                 0×00000000
                                                                                                                 0×12345004
         rs1_val
             rs2
         rs2_val
                                0×12345004
                                                                                                                 0×12345004
       alu_a_sel
                                                                               rs1
       alu_b_sel
                             imm_x
        alu_func
                                 Q=A+B
                                                                                                  Q=A+B
                                                                                                                  Q=A+B
         sub_sra
                                                0×000000008
                                                                                 0×00000030
           \mathtt{imm}_{-} \mathtt{x}
                                0×000000008
                                                                0×000000008
                                                                                                 0×00000020
                                                                                                                 0×00000020
              alu
         mem_clk
        addr_sel
            addr
                                                        0×00000020
                                                                                         0×00000030
         mem_out
                                                                                         0×02f00063
                                                                                                         0×03010063
                                                        0x008000et
                                                                         0×03000267
 sx\_size[2..0]
          mem_sx
                                                                                                           t_{11}
                                                                                                                   t_{12}
                                                                                                                            t_{13}
```

Figure 4: Fetch, decode, and execute of load, store, jump, and branch instructions.