Solution

Problem 1: (12 points)

1. AB or BA

2. Two processes. The first Fork() returns 0 in the first child process and the second Fork() will not be executed in it. But in the parent process, the first Fork() returns a non-zero PID so the second one will be executed and the second child process is created

3. Two X. waitpid(-1, NULL, 0) will return two times for the two processes.

Problem 2: (16 points)

1. In parent: 1, in child: 3

   or

   In parent: 1, in child: 2

2. c:16

   p:-7

   or

   c:99

   p:-7

3. kill –USR1 1000

   kill –USR1 1000

   kill –USR1 1000

   kill –USR2 1000

   kill –USR2 1000

   kill –INT 1000

4. The n will be negative, which may result in segmentation fault and program crash. Register a handler for SIGSEGV to handle segmentation fault.

Problem 3: (27 points)

1. [1] valP ← PC + 1

   [2] valA ← R[%eax]

[3] valB ← R[%ebx]

[4] valE ← valA + valB

[5] valM ← M4[valE]

[6] R[%eax] ← valM

([2]和[3]里的寄存器编号可以对换，但是必须和 Part B 中的 HCL 代码一致)

2. int srcA = [

    icode in { IRRMOVL, IRMMOVL, IOPL, IPUSHL } : rA;

    icode in { IPOPL, IRET } : RESP;

    icode in { IXLAT } : REAX;

    1 : RNONE;

];

int srcB = [

    icode in { IOPL, IRMMOVL, IMRMOVL } : rB;

    icode in { IPUSHL, IPOPL, ICALL, IRET } : RESP;

    icode in { IXLAT } : REBX;

    1 : RNONE;

];

int dstM = [

    icode in { IMRMOVL, IPOPL } : rA;

    icode in { IXLAT } : REAX;

    1 : RNONE;

];

(srcA 和 srcB 的实现必须和 Part A 中[2]和[3]一致，icode in { IXLAT } 也可以写成 icode == IXLAT)

3. Other circuit logics that are required to be modified are instr_valid, aluA, aluB, mem_read and mem_addr.

(送 1 分，5 个信号每个 1，其它信号每多写一个扣 1)

Problem 4: (20 points)

1 [1] 0x21 [2] 0x1c [3] 0x21 [4] 0xd

[5] 0xd [6] 0x4 [7] 0x0 [8] 0x8

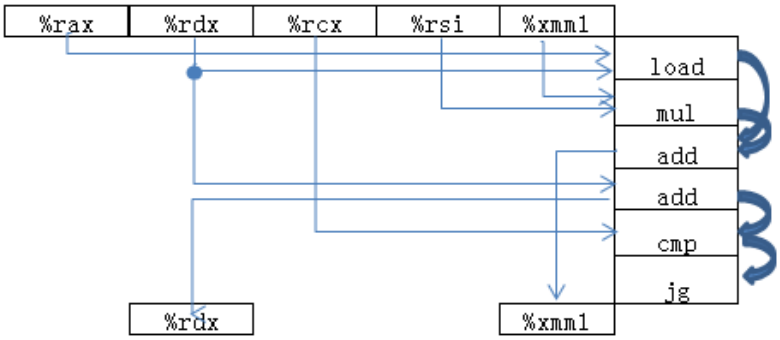
[9] 0x4 [10] 0x3

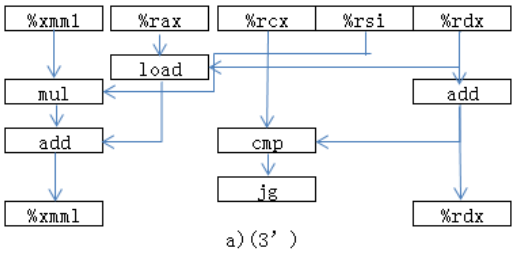
Problem 5: (12 points)

```
void foo(vec_ptr v, data_t *dest)
{
   data_t preSum = 0;                    ← [1]
   int length = get_vec_length(v);       ← [2]
   data_t temp = 0;                      ← [3]
   for (int i = 0; i < length; i++) {    ← [4]
      preSum *= v->data[i] << 1;         ← [5]
      temp += v->data[i-1];              ← [6]
      v->data[i-1] = v->data[i];
      v->data[i] = temp;
   }
   *dest = preSum;
}
```

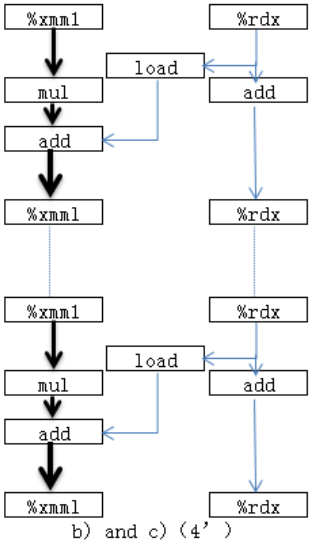
[1] used for eliminate unneeded memory references (2’)
[2] reducing procedure calls (2’)
[3] local variable for expansion of function (2’)
[4] loop fusion (2’)
[5] replace multiplication with shifting (2’)
[6] expansion of function in loop (2’)

Problem 6: (13 points)





a)(3')



b) and c) (4')

a) 4' b) 3' c) 2' d) 2' critical path: 2'