ECEN 651   HW2   Zhenlei Song

1. Effective CPI $= \Sigma$ mix frequency $\times$ clock cycle

(1) ~~ALU~~ Load / store

load	26.5%	25.1%	25.8% ave
store	10.3%	13.2%	11.8%

~~(2)~~ instruction mix freq $= 25.8\% + 11.8\% = 0.376$

(2) conditional branch:

cond branch   9.3%   12.1%   10.7% ave

instruction mix freq $= 0.107$

(3) jumps

return	1.6%	0.6%	1.1%
call	1.6%	0.6%	1.1% ave
jump	0.8%	0.7%	0.8%

instruction mix freq $= 3\% = 0.03$

(4) ALU

all others will be considered as ALU

add	21.1	19.0	20	cond move	0.4	0.6	0.5
sub	1.7	2.2	2.0	compare	2.8	6.1	4.4
mul	1.4	0.1	0.8	other logic	0.1	0.4	0.2
or	7.9	8.5	8.2				
xor	1.8	2.1	2.0	instruction mix freq			
and	4.3	4.6	4.4	$= 0.485$			
shift	3.8	1.1	2.4				
load imm	4.8	2.5	3.6				

Continued

conditional branches clock cycle = $((0.6 \times 2.0) + (1 - 0.6) \times 1.5) = 1.8$

Effective CPI =

$(0.485 \times 1.0) + (0.376 \times 1.4) + (0.107 \times 1.8) + (0.03 \times 1.2)$

= 1.24

2.

A.7(a)

```
         DADD    R1, R0
         SW      R1, 7000 (R0);
Loop:    LD  R1, 7000 (R0);
         DSLL· R2, R1, 3;              2 + (16 × 101) = 1618 instructions
         DADDI   R3, R2, 3000;
         LD      R4, 0(R3);            8 × 101    4/s  instructions
         LD      R5, 5000 (R0);
         DADD    R6, R4, R5;           code size :
         LD      R1, 7000 (R0);        4B × 18 = 72B
         DSLL    R2, R1, 3;
         DADDI   R7, R2, # 1000;
         SD      R6, 1000 (R7);
         LD      R1, 7000 (R0);
         DADDI   R1, R1, #1;
         SD      R1, 7000 (R0);
         LD      R1, 7000 (R0);
         DADDI   R8, R1, #-101;
         BNEZ    R8, loop;
```

3.

A.20 a

ALU   length   =   16 bits

~~branch~~ type   :   ~~16~~ 16 bit   ratio = 0.304

data ref   24   bit   ratio   = 0.669 - 0.304 = 0.365

                       32   bit   ratio   = 1 - 0.669

branches type :   = 0.331

                 16 bit   ratio   = 0.001

                 24 bit   ratio   = 0.852 - 0.001 = 0.851

                 32 bit   ratio   = 1 - 0.852 = 0.148

data ref length $= 16 \times 0.304 + 24 \times 0.365 + 32 \times 0.331$

                     = 24.2 bit

branches   length   = $16 \times 0.001 + 24 \times 0.851 + 32 \times 0.148$

                     = 25.3 bit

ALU   freq   =   add (19%) + sub (3%) + compare (5%)

              + cond move (1%) + shift (2%) + and (4%)

              + or (9%) + xor (3%)   + 1%

              = 47% = 0.47

data ref   = load (26%) + store (10%) + load immm (2%)

              = 38% = 0.38

branch freq = 1 - 0.38 - 0.47 = 0.15

average length   = $16 \times 0.47 + 24.2 \times 0.38 + 25.3 \times 0.15$

                 = 20.5 bits