Vitaliy Prymak

Homework 2

1. We wish to compare the perfomance of two different machines: M1 and M2. The following measurements have been made on these machines

Program	Time on M1	Time on M2
1	3 seconds	1,5 seconds
2	4 seconds	10 seconds

a) Which computer is faster for each program and how many times as fast is it?

Perfomance $= \frac{1}{\text{Execution Time}}$; Relative perfomance $= \frac{\text{Perfomance } M1}{\text{Perfomance } M2}$

Program 1: $\frac{\text{Execution Time } M2}{\text{Execution Time } M1} = \frac{1.5}{3} = 0.5$. So M2 0.5 faster

Program 2: $\frac{10s}{4s} = 2.5$. SO M1 2.5 times faster M2

b) If the following additional measurements were made:

Program	Instructions executed on M1	Instructions executed on M2
1	5×10^9	6×10^9

and the clock rates of machine M1 and M2 are 2.5 GHZ and 2 GHz respectively, find the clock cycles per instruction (CPI) for program 1 on each computer

M1: Clock Rate = Clock Cycle Time = 2.5 GHz ; $IC = 5 \cdot 10^9$

CPU Clock Cycles = $IC \cdot CPI$

CPU Time = CPU clock Cycles $\cdot$ CCT ; CPU Time = 3S

CPU Time $= \frac{IC \cdot CPI}{CR}$; $3S = \frac{5 \cdot 10^9 \cdot CPI}{2.5 \cdot 10^9}$ $2 = \frac{6}{3}$

$CPI = \frac{3 \cdot 2.5 \cdot 10^9}{5 \cdot 10^9} = \frac{3 \cdot 2.5}{5} = \frac{7.5}{5} = 1.5$; $CPI = 1.5$

M2: $IC = 6 \cdot 10^9$; $CR = 2GHz$, CPU Time = 1.5S ; $CPI = 0.5$

c) Assuming that CPI for program 2 on each computer is the same as CPI for program 1 found in b, find the instruction count for program 2 running on each computer using the execution times from the first table.

$$CPU\ Time = CPU\ CC \cdot \frac{1}{RC} = \frac{IC \cdot CPI}{RC}$$

$CPI_{1,2} = 1,5; 0,5;$

$CPU\ Time_{1,2} = 4s,\ 10s$; $CR_{1,2} = 2,5\ GHz,\ 2\ GHz$

$IC_{1,2}$ - ?

M1: $4 = \dfrac{IC \cdot 1,5}{2,5 \cdot 10^9}$

$1,5 \cdot IC = 4 \cdot 2,5 \cdot 10^9$

$IC = \dfrac{4 \cdot 2,5 \cdot 10^9}{1,5}$

$IC_1 = \dfrac{10 \cdot 10^9}{1,5} = 6,7 \cdot 10^9$

M2: $10 = \dfrac{IC \cdot 0,5}{2 \cdot 10^9} = IC \cdot 0,5 = 10 \cdot 2 \cdot 10^9$

$= \dfrac{20 \cdot 10^9}{0,5} = 40 \cdot 10^9$

2. Consider three different processors P1, P2, P3 executing the same instructions set. P1 has a 3 GHZ clock rate and a CPI of 1,5. P2 has a 2.5 GHZ clock rate and a CPI of 1.0. P3 has a 4.0GHz Clock Rate and has a CPI of 2.2.

ⓐ Which processor has the highest perfomance expressed in instructions per second?

ⓑ If the processor P1 execute a program in 10 secoands, find number of cycles and the number of instructions.

ⓒ We are trying to reduce the execution time by 20% for P1, but this leads to an increase of 30% in the CPI. What clock rate should we have to get this time reduction for P1?

a) Processor P_1 : $CR = 3 \cdot 10^9$, $CPI = 1,5$

P_2 : $CR = 2,5 \cdot 10^9$, $CPI = 1,0$

IC - ? P_3 : $CR = 4,0 \cdot 10^9$, $CPI = 2.2$

~~$CPU\ time = \frac{IC \cdot CPI}{CR}$~~

Perfomance (in instructions per second) =

$= \frac{\text{Clock Rate (in Hertz)}}{\text{CPI (Cycles Per Instruction)}}$

Perfomance represents number of instructions processor can execute in one secoand.

Clock Rate is speed of processor in Hertz

CPI - how many cycle clock needed to execute single ins

$P_1 = \frac{3 \cdot 10^9}{1,5} = 2 \cdot 10^9 = 2{,}000{,}000{,}000$ instructions per second

$P_2 = \frac{2.5 \cdot 10^9}{1} = 2.5 \cdot 10^9$ hihgest perfomance.

$P_3 = \frac{4 \cdot 10^9}{2.2} = 1,8 \cdot 10^9$

b) CPU Time P_1 = 10s ; CR_{P_1} = 3 GHz

CPI, IC - ?

$\cancel{\text{CPU Time} = \frac{IC \cdot CPI}{CR}}$

Number of Cycles = Clock Rate · Time (s) =

= 3 GHz · 10s = 30,000,000,000 cycles

Number of Instructions = $\frac{\text{Number of Cycles}}{\text{Cycles Per Instruction}}$

$= \frac{3\,GHz}{1,5}$ = 20,000,000,000 instructions.

So if processor P1 executes program in 10 sec. it would perform $30 \cdot 10^9$ cycles and executes $20 \cdot 10^9$ instructions during that time.

c) $\text{CPU Time} = \frac{IC \cdot CPI}{CR} \Rightarrow CR = \frac{IC \cdot CPI}{\text{CPU Time}}$;

$= \frac{IC \cdot CPI \cdot \frac{130}{100}}{\text{CPU Time} \cdot \frac{80}{100}} = \frac{IC \cdot CPI \cdot 1,3}{\text{CPU Time} \cdot 0,8} =$

! 1,3 represent 30% increase

! 20% decreased

$= \frac{\cancel{3}\ 20 \cdot 10^9 \cdot 1,5 \cdot 1,3}{10 \cdot 0,8} = 4,875\ GHz$

3. Consider two different implementatio[ns] of the same instruction set architectu[re]. The instructions can be divided into fo[ur] classes according to their CPI (classes A, B, C, D). P1 with clock rate of 3GHz and CPIs of 1, 2, 3 and 3, and P2 with clock of rate of 2GHz and CPIs of 2, 2, 2 and 2. Given a program with a dynamic instruction count of 10^3 instructions divided into classes as follows: 50% class A, 20% class B, 20% class [C] and 10% class D.

a) What is the global CPI for each implementation?

b) Which is faster: P1 or P_2?

a) $CPI = \sum_{i=1}^{n} CPI_n + \text{Frequency}$:

Classes	%	P1 CPI	P2 CPI	P1CR
A	50%	1	2	3GHz
B	20%	2	2	3
C	20%	3	2	3
D	10%	3	2	3

$CPI_{P_1} = 1 \cdot \underset{0,5}{50\%} + 2 \cdot \underset{0.2}{20\%} + 3 \cdot \underset{0.2}{20\%} + \underset{0.1}{10\%} \cdot 3 = 1.8$

$CPI_{P_2} = 2 \cdot 50\% + 2 \cdot 20\% + 2 \cdot 20\% + 2 \cdot 10\% = 2$

b) $CPU Time\ P_1 = \frac{IC \cdot CPI_1}{CR} = \frac{10^3 \cdot 1,8}{3 \cdot 10^9} = \frac{1,8 \cdot 10^3}{3 \cdot 10^9} = 0,6 \cdot 10^{-6}$

If in denominator 10^6 if we put in numerator 10^{-6}

$CPU Time\ P_2 = \frac{IC \cdot CPI_2}{CR} = \frac{10^3 \cdot 2}{2 \cdot 10^9} = 1 \cdot 10^{-6}\,s$

P_1 will be faster because it has lower executi[on]

4 Its often needed to exchange the contents of two registers. It takes three instructions to accomplish this in LEGv8. It might be useful to add a new instruction to LEGv8 ISA that take two registers and exchanges their contents. Suppose that CPI is the same across all instruction types, and moreover is unaffected by the new instruction, but that the clock rate decreases by 15% Suppose that addition of this new instruction also reduces the total instruction count by 9%. Is it beneficial to implement this enhancement? If yes, what is overall speedup factor obtained?

! ~~CR · 0,15~~, ~~IC · 0.09~~, ~~CPI is same~~

New: CR · 0,85, IC · 0,91

Old: CR · 1, IC · 1

$$\frac{0,91}{0,85} = 1,1$$

New > Old 1,1 > 1,0

It's not beneficial because it's large than old one.

5. Assume that computer M1 has CPU clock rate 2GHz and supports four classes of instructions: ALU, load, store, branch/jump. Load has CPI = 5, store has CPI = 4, ALU instruction has CPI = 4, and branch/jump has CPI = 3. The following program P is used in testing the computer perfomance:

```
F: SUBI SP, SP, #16 // (*) make room on the stack
STUR X19, [SP, #0] // preserve X19
STUR X30, [SP, #8] // preserve the return location
ADD X19, X2, X3 // calculate c+d
 // Clean up in preparation for tail call
LDUR X19, [SP, #0]
LDUR X30, [SP, #8]
LSL X30, X30, #2
ORR X30, X30, X19
ADDI SP, SP, #16 // (*)
 B F // call g(g(a,b), c+d) with tail-call optimization
```

a) What is the average CPI for program P?

b) Another computer M2 with clock rate of 3GHz has the same perfomance as computer M1 when it runs the program P. What is the average CPI of computer M2?

	CPI	CR	IC
ALU	4	2	~~5~~ 3
Load	5	2	2
Store	4	2	2
branch/jump	3	2	1
			8 ← Total Instructions

!

~~IC = 16~~

$$\cancel{CPI = \sum_{n=1}^{4} CPI_n \times Frequency =}$$

$$\cancel{= 4 \cdot 2 + 5 \cdot 2 + 4 \cdot 2 + 3 \cdot 2 = 32 \cdot 10^9}$$

① First need to count how many each instructions from code: IC

$$CPI_{ave} = \frac{CPI_{\text{each Instru}} \cdot IC}{\text{Total Instru}} = \frac{4 \cdot 3 + 5 \cdot 2 + 4 \cdot 2 + 3 \cdot 1}{8}$$

$$= \frac{12 + 10 + 8 + 3}{8} = \frac{33}{8} = 4,14$$

② Perfomance $= \frac{CR\,M2}{CR\,M1} = \frac{3 \cdot 10^9}{2 \cdot 10^9} = 1,5.$

To achieve the same perfomance, the CPI for each instruction on M2 would need to scale down by same factor

$CPI_{ALU} = 4/1,5 = 2,67,$ $CPI_{Load} = 5/1,5 = 3,33$

$CPI_{store} = 4/1,5 = 2,67,$ $CPI_{branch} = 3/1,5 = 2$

$$CPI_{M2\,average} = \frac{2,67 \cdot 3 + 3,33 \cdot 2 + 2,67 \cdot 2 + 2 \cdot 1}{8} =$$

$$= \frac{8 + 6,7 + 5,3 + 2}{8} = \frac{22}{8} = 2,75$$

6. A pitfall is expecting to improve overall performance of a computer by improving only one aspect of the computer. Consider a computer running a program that requires 250s, with 70s spent executing FP instructions, 55 s executed ALU Instructions, 85s executed load/store instructions, and 40 s spent executing branches instructions.

a) By how much is the total time reduced if the time for FP operations is reduced by 20%?

b) Can the total time can be reduced by 20% by reducing only the time for branch instructions? Why?

a) Program CPU Time = 250s

FP = ~~55~~ 70 s, + ALU = 55 s, + load = 85 s, + branches 40 s = 250 s

$$T_{improvement} = \frac{T_{affected}}{T_{improvement\ amount}} + T_{uneffected} =$$

70 % 20 = 14

70 − 14

$$= \cancel{\frac{70}{1,2}} + 180 = 56 + 180 = 238s$$ Time reduced from 250 s to 238s by reducing time of F

b) $T_{imp} = \cancel{\frac{40}{1,99}} + 210 s.$ ~~8~~ 20

$= \frac{250}{} \% 20 = 50$

$250 - 50 = 200s$

and branch instruction execution time is
even if we reduce branch execution
to 0 total exec will be 210s but
20% is = to 200 therefore won't reduce by 20

7. Given the 32-bit binary number

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
1101	0011	0110	0000	0001	1100	1010	1111

What does it represent respectively, assuming that it is

a) a signed (i.e. already in 2's compliment) integer? (converted in decimal)

b) unsigned Integer? (convert to decimal)

c) a single precision floating point number? (write in scientific notation)

d) a LEGv8 instruction?

a) Signed $(2^{31}+2^{30}+2^{28}+2^{25}+2^{24}+2^{22}+2^{21}+2^{12}+2^{11}+2^{10}+2^{7}+2^{5}+2^{3}+2^{2}+2^{1}+2^{0}) = 2,147,483,648 + 1,073,741,824 + 268,435,456 + 33,554,432 + 16,777,216 + 4,194,304 + 2,097,152 + 4096 + 2048 + 1024 + 128 + 32 + 8 + 4 + 2 + 1 = -3,546,291,425$

b) Unsigned: 3,546,291,425

c) 1, 10100110110 0000 000111 001010 1111 $\cdot 2^{-30}$

d)

11 bits	5 bits	6 bits	5 bits	5 bits
1, 1010011011	0 0000	000111	00101	0 1111
OP code LSL	Second Register	Shift 7	First Regis 5	Destin Regis 15

R-TYPE LSL X15, X5, #7

8. With x =
0000 0000 0000 0000 0000 0000 0101 1011
and y =
0000 0000 0000 0000 0000 0000 0000 1101
representing two signed integers, perform, showing all work.

a. x + y

b. x − y

c. x · y

a)
```
  0000 0000 0000 0000 0000 0000 0101 1011
+ 0000 0000 0000 0000 0000 0000 0000 1101
  0000 0000 0000 0000 0000 0000 0110 1000
```

b) 0000 0000 0000 0000 0000 0000 0100 1110

c)
```
     01011011
   + 00001101
     01011011
    00000000
   01011011
  01011011
0........ 10010011111
```

9. Show the IEEE 754 binary representation for the floating point number 20.125 and -5.75 in a single and double precision

$20,125 = 10100.001 =$

$= 10100.001 \cdot 2^0 =$

$= 1,\underbrace{0100001}_{\text{fraction}} \cdot 2^4$ ← shifted exponent

$10100.$ (16, 4)

$0,125 \times 2 = 0,250$

$0,250 \times 2 = 0,500$

$0,500 \times 2 = 1,000$

exponent = shifted exponent $+127 = +131$

10000011 (128, 2, 1)

sign (1)	8 bits	23
0	10000011	0100001...0

32 bits

Single point precision

Double point precision: exponent = shifted expon + bias

$= 4 + 1023 = 1027$ $\quad 10000000011$ (1024, 2, 1)

1	11 bits	52 bits
0	10000000011	0100001000... 0

64 bits

$-5,75 = -101.11 = -101.11 \cdot 2^0 = -1,0111 \cdot 2^2$

$101.$

$0,75 \times 2 = 1,50$

$0,50 \times 2 = 1,00$

Exponent = Shifted exponent + bias

$= 2 + 127 = 129$

1 bit	8 bits	23 bits
1	10000010	0111 ... 0

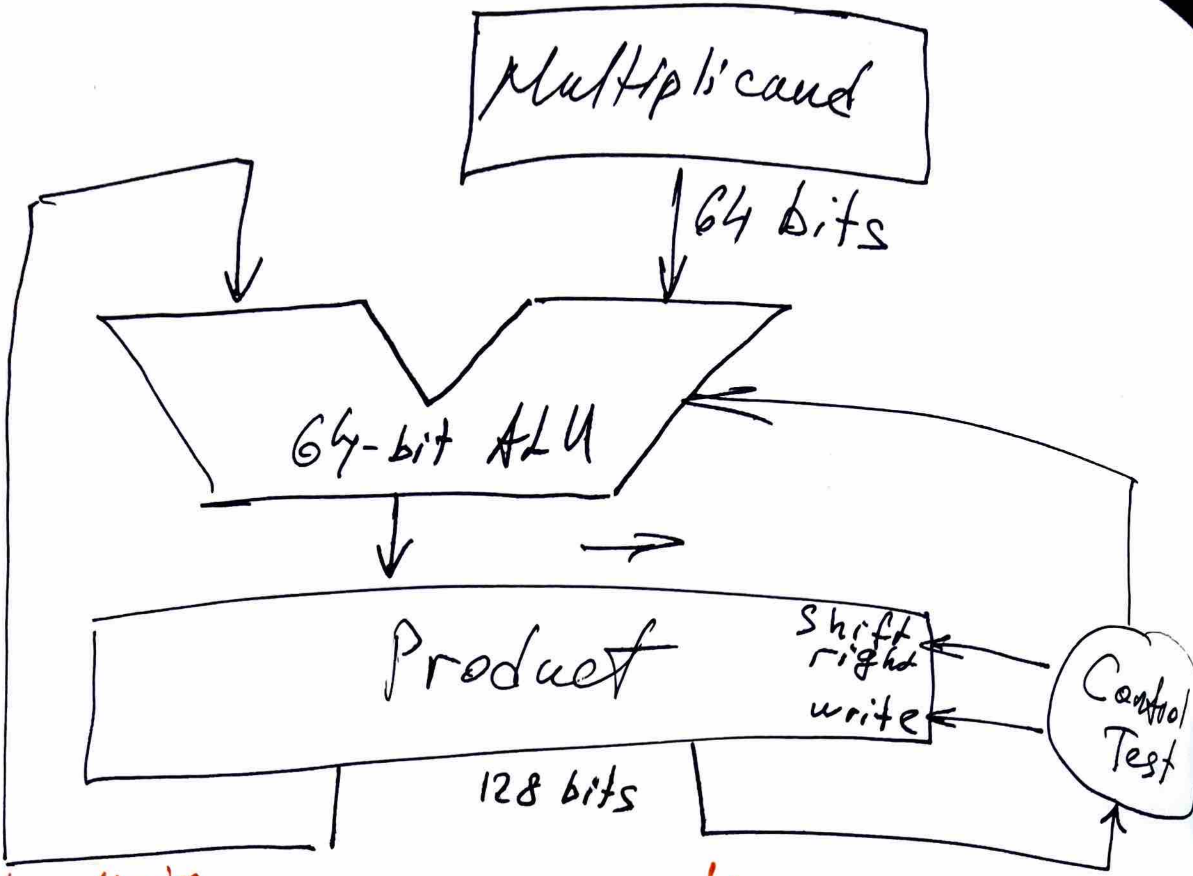
Double precision Exponent = Shifted Exponent + bias $= 2 + 1023 = 1025$

1 bit	11 bits	52 bits
1	10000000001	0111000 ... 0

10. Perform (20.125) + (-5.75) and (20.125) · (-5.75) in binary normalized scientific format obtained from Question 9. Remember to normalize the result.

	Operation	Multiplicand	Product
0	Initial value (load multiplier to the lower half of the product register)	1010	0000 0101
1	Add: Prod = Prod + Mcand	1010	0000 0101 +1010 1010 0101
	Shift: Product to the right	1010	0101 0010
2	Add: No adding	1010	0101 0010
	Shift: shifting	1010	0010 1001
3	Add: Adding = Prod + Mcand	1010	0010 1001 1010 1100 1001
	Shift: Shifting to right	1010	0110 0100
4	Add: No adding	1010	0110 0100
	Shift: Shifting		0011 0010

Multiply 1010 by 0101 using the optimized multiplier. For each step show the contents of the product register and the operation using following table.



① Normalizing

$20,125 = 1,0100001 \cdot 2^4$

$\pm\ 5,75 = 1,0111 \cdot 2^2$

② Chosing smallest exponent $1,0111 \cdot 2^2$ and converting to largest $0,010111 \cdot 2^4$

$$\begin{array}{r} \pm\ 1,0100001 \\ 0,010111 \\ \hline 0,1110011 \end{array} \cdot 2^4$$

back to 2^2 = 11,10011

Multiplication

$$\begin{array}{r} 1,0100001 \cdot 2^4 \\ \times\ 1,0111 \cdot 2^2 \end{array}$$

① Adding exponents

$4 + 2 = 6$

② Perform multiplication

$$\begin{array}{r} 1,0100001 \\ 1,0111 \\ \hline 10100001 \\ 10100001 \\ 10100001 \\ 00000000 \\ 10100001 \\ \hline -\ 1,11001110111 \end{array} \cdot 2^6$$

1	1000000101	11001110111

$6 + 127 = 133$

11. Given the following 1-bit ALU, and 3-bit control codes (Ainvert, Operation), write the operations performed corresponding to the 8 control codes in the following table

Ainvert, Operation		Operation performed Result =
0	00	~~[illegible]~~ A + B
0	01	A || B
0	10	A AND B
0	11	~~shift~~ less (jump)
1	00	A' + B
1	01	A' || B
1	10	A' & B
1	11	Less'