3nd Homework for Computer Architecture

Submission deadline: Dec.6 ， 11:59pm
（total 120 points）

B.1 [12] Compute the effective CPI the following processor. Suppose we have made the following measurements of average CPI and mix frequency for instructions:

Instruction	Frequency	Clock cycles
All ALU operations	43%	1.0
Loads	21%	2.0
Stores	12%	1.6
Conditional Branches	23%	
Taken	60%	2.0
Not taken	40%	1.5
Jumps	1%	1.2

solution：

计算处理器的有效 CPI

*0.43 * 1 + 0.21 * 2 + 0.12 * 1.6 + 0.23 * 0.6 * 2 + 0.23 * 0.6 * 1.5 + 0.01 * 1.2 = 0.43 + 0.42 + 0.192 + 0.276 + 0.138 + 0.012 = 1.468*

B.2 [20/15] Consider adding a new index addressing mode to DLX. The addressing mode adds two registers and an 11-bit signed offset to get the effective address. Our compiler will be changed so that code sequences of the form

```
ADD     R1,   R1,   R2
LW      Rd, 100(R1)   (or store)
```

Will be replaced with a load (or store) using the new addressing mode. The new instruction might be in the following format:

Opcode(6b) Rs(5b) Rt(5b) Rd(5b) offset(11b)　　Rd ←MEM[(Rs)+(Rt)+ offset]

Use the overall average instruction frequencies from the figure in above B.1 in evaluating this addition.

DLX 的新的寻址模式。这个寻址模式将两个寄存器的值以及 11 位的有符号偏移量求和计算有效地址。我们的编译器会改变，将下面的 add 和 lw 合并，为新的寻址模式。新的指令的格式如下。

a. [20] Assume that the addressing mode can be used for 10% of the displacement loads and stores (accounting for both the frequency of this type of address calculation and the shorter offset). What is the ratio of instruction count on the enhanced processor compared to the original processor in B.1?

假设这种寻址模式可以用于 10%的偏移量 load 和 store。计算指令数减少了多少

solution：

*(21+12) * 0.1 = 3.3, 也就是说，比原来减少了3.3%的指令数。注意少的指令数是 alu 运算的指令数*

b. [15] If the new addressing mode lengthens the clock cycle by 5%, which machine will be faster and by how much?
新的寻址模式提高了 5%的时钟周期，哪个更快。

solution：

*新的 lw 时钟周期 = 2 * 1.05 = 2.1*

*新的 sw 始终周期 = 1.6 * 1.05 = 1.68*

*因此新的 CPI = 1.468 + (2.1-2) * 0.21*0.1 + (1.68-1.6)*0.12*0.1 – 0.033 * 1 = 1.468 + 0.0021 + 0.00096 – 0.033 = 1.3806*

因此第二个更快，提高了 (1.468-1.3806) / 1.468 = 5.95%

B.3 [18] When designing memory systems it becomes useful to know the frequency of memory reads versus writes and also accesses for instructions versus data. Using the average instruction-mix information for processor in B.1, find

- The percentage of all memory accesses for data.
- The percentage of data access that are reads.
- The percentage of all memory accesses that are reads.

Ignore the size of a datum when counting accesses.

solution：

内存数据访问比率. //这里的比率的总数是指的所有的内存访问次数
(0.21 + 0.12) / (1 + 0.21 + 0.12) = 25%
内存读数据比率
0.21 / 1.33 = 16%
所有读内存的比率
(1+0.21) / 1.33 = 91%

B.4 [5/15/15]Suppose you are designing a new laptop computer for running high thread-count applications, and you want to decide which of three different processor chip types to incorporate into the system design:

- Chip A is a superscalar uniprocessor costing $200 with a clock frequency of 3.2 GHz. Its average CPI on an appropriate benchmark is 0.6, and it dissipates 15 Watts of power.

- Chip B has the same instruction set as chip A, costs $900, and is a dual-core processor with 2 hyperthreaded processors, each supporting 2 virtual CPUs. Its clock frequency is 3.6 GHz. Its average CPI on the benchmark (which is multithreaded) is 0.4. It consumes 30 W of power.

- Chip C is a pipelined RISC-style processor costing $100 with a clock speed of 2.5 GHz. Its average CPI on the benchmark is 1, and it uses 10 W. However, its instruction count on the benchmark is twice that of chips A and B.

Suppose your product manager tells you that the laptop must cost no more than $1,000 altogether (including all parts) and must dissipate no more than 30 W of power in total, including the display, hard drive, etc. Assume that the parts other than the CPU have already been selected, and that together they cost $300 and burn a total of 5 W of power.

(a) [5]*Identify* the engineering problem to be solved. In the scenario described, what should you, as the system architect, be trying to do? Select only one:

(i) Always just pick the chip with the highest clock speed, and never look back.
(ii) Calculate the MIPS rating of each chip, and select the one with the highest MIPS rating.
(iii) Calculate the relative execution time of each chip on the benchmark, and select the one with the lowest execution time.
(iv) Calculate the cost-performance (performance per unit cost) of each chip, and select the one with the best cost-performance.
(v) For each chip type, calculate the maximum total throughput that can be achieved within the design constraints by using multiple instances of the chip taken together, and select the best-performing resulting design.

solution：*必然选择的是(V)。其他的任何策略都不能保证最好的性价比。*

(b) *Formulate* the engineering problem, by composing an algebraic expression for the figure of merit that you identified in part (a), in terms of the following variables:

$¢_{chip}$ – Cost per processor chip of a given type.
$¢_{sys,max}$ – Maximum cost of the entire system.
$¢_{aux}$ – Cost of all auxilliary (non-processor) components taken together.
f_{clk} – Clock frequency of a chip of a given type.

CPI_{chip} – Average CPI of a chip of a given type on the benchmark.

P_{chip} – Power dissipated per chip of a given type.

$P_{sys,max}$ – Maximum power dissipation of the entire system.

P_{aux} – Power dissipated by all auxilliary components taken together.

IC_{rel} – Relative instruction count for the ISA used by a given chip technology, expressed as a multiple of the IC for Chip A.

solution：

设计中用到的在给定类型下最多数量的芯片

$$n_{chips} = \text{floor}(\min[(¢_{sys,max} - ¢_{aux})/\ ¢_{chip,}(\ P_{sys,max} - P_{aux}\ )/\ P_{chip}])$$

最大数量芯片下的相对吞吐量

$$T_{rel} = n_{chips} \times f_{clk} / (IC_{rel} \times CPI_{chip})$$

(c) *Solve* the engineering problem by plugging in the numbers from the problem description into your formula from part (b) for each processor type, evaluating the figure of merit that you identified in part (a), and showing which of the three processor types gives the best resulting value for that figure of merit.

solution：

对于所有的情况

$¢_{sys,max} - ¢_{aux} = \$700$

$P_{sys,max} - P_{aux} = 25W$

chip A:

$n_{chips} = \text{floor}(\min[(\$700)/\ \$200, 25W/15W]) = \text{floor}(\min[(3.5, 1.67) = 1$

$T_{rel} = 1 \times 3.2GHz / (1 \times 0.6) = 5.333 \times 10^9$

chip B:

$n_{chips} = \text{floor}(\min[(\$700)/\ \$900, 25W/30W]) = \text{floor}(\min[(0.77, 0.83) = 0$

$T_{rel} = 0 \times 3.6GHz / (1 \times 0.4) = 0$

chip C:

$n_{chips} = \text{floor}(\min[(\$700)/\ \$100, 25W/10W]) = \text{floor}(\min[(7, 2.5) = 2$

$T_{rel} = 2 \times 2.5GHz / (2 \times 1) = 2.5 \times 10^9$

因此，chip A 拥有最好的相对总体性能—也就是相对总吞吐量。
根据上面描述的方法，chip A 是我们的选择。

B.5 (20 points) Compare the differences of RISC and CISC.

solution：

- In RISC, the CPU control can be done with hardwired without comprising a control memory whereas CISC is micro coded that uses ROM, However, the current CISC processor also utilizes hardwired control
- RISC processor works with 32-bits for each instruction and frequently based on the register while CISC utilizes an uneven format that ranges from 16 bits to 64 bits for each inst.
- RISC architecture includes the design of instruction cache and split data whereas CISC architecture includes a unified cache intended for data & instructions, even though most recent design also utilize split caches
- in the RISC processor, the single clock is used, and addressing modes are limited whereas, in CISC, it uses the multi clock, and addressing modes range from 12 to 24.
- RISC ISA highlights software as compared with hardware. The instruction set of the RISC processor uses more efficient software like code or compilers through fewer instructions. CISC ISAs employ a number of trnasistors within the hardware to execute several instructions as well as additional complex instructions also

RISC	CISC
RISC stands for reduced instruction set computer	CISC stands for Complex instruction set computer
RISC processors have simple instruction taking about one clock cycle. the average Cpi is 1.5	CISC processors have complex instruction set that take up multiple clocks for execution. the average Cpi is the range of 2 and 15
Performance is optimized with more focus on software	Performance is optimized with more focus on hardware
it has hard-wired unit of programming	It has a microprogramming unit
It only has a few instructions	A variety of different instructions
RISC processors are highly pipelined	Less pipelined
Execution time is very less	Execution time is very high
Code expansion can be a problem	Code expansion is not a problem
The decoding of instructions is simple	Decoding of instructions is complex
Etc..	Etc..

RISC 和 CISC 设计的 philosophy 就是不同的，因此其实还有更多的不同点，这里列举了主要的重要的以及自己理解的一些区别。
reference：https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/risccisc/