CENG3420 Homework 2

Due: Mar. 29, 2017

Solutions

Q1 (**15%**) You are required to develop some simple measures of pipeline performance and relative speedup.

1. Let $T_{k,n}$ be the total time required for a pipeline with k stages to execute n instructions. Speedup of k stage pipeline is given by,

$$S_k = \frac{T_{1,n}}{T_{k,n}}. \tag{1}$$

   Determine S_k in terms of k and n.

2. Consider an instruction sequence of length n that is streaming through the instruction pipeline. Let p be the probability of encountering a conditional or unconditional branch instruction, and let q be the probability that execution of a branch instruction I causes a jump to a nonconsecutive address. Assume that each such jump requires the pipeline to be cleared, destroying all ongoing instruction processing, when I emerges from the last stage. Determine S_k in terms of k, n, p and q.

A1 1.

$$S_k = \frac{nk}{k+n-1}.$$

2.

$$S_k = \frac{nk}{pqnk + (1-pq)(k+n-1)}.$$

Q2 (**15%**) Considering the single-cycle datapath shown in Fig. 1, you are required to redesign it into a multicycle path.

1. How many state registers are required? Where should those registers be placed? What are the functions of them?
2. In which stage control signals are generated for the above multi-cycle processor.

A2 1. We need to use four inter-stage registers which should be inserted after Instruction Memory, Registers, ALU and Data Memory. They are used to store the intermediate value and control signal to implement pipeline.

2. ID stage.

Q3 (**15%**) Answer the following questions about 4-bit muliplication implementation of ALU design:

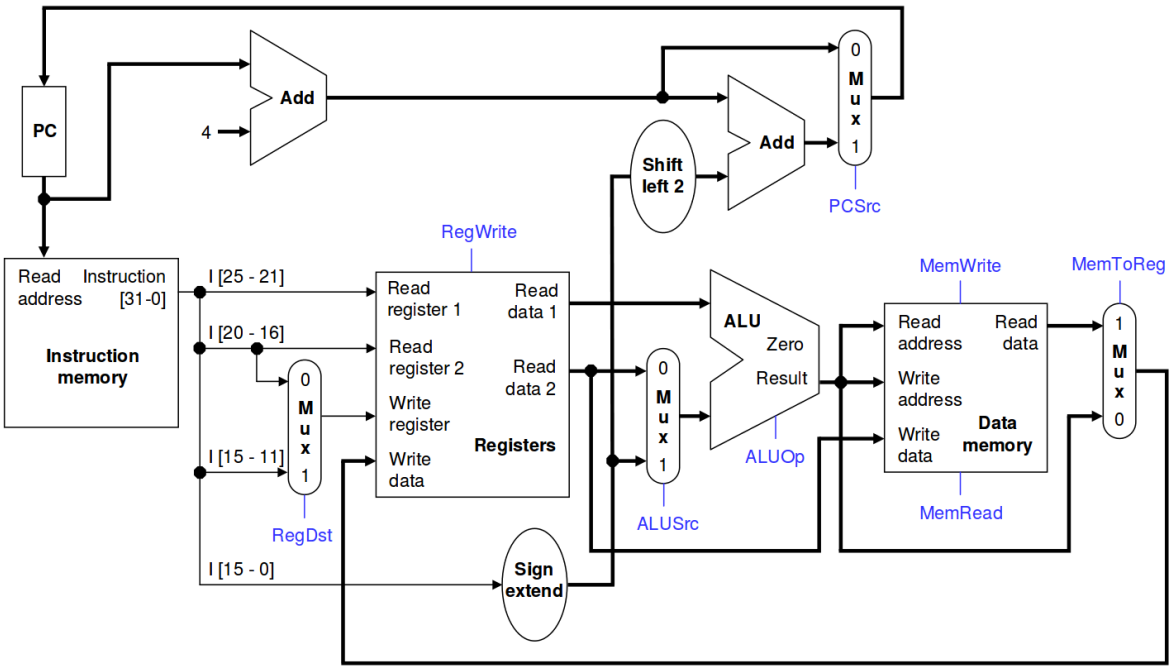

Figure 1: Q2

1. A straightforward way to do multiplication is shifting and adding. Write down the details of each iteration on calculating 5×6.
2. If we want to calculate 5×-6, the above method will fail beacuse -6 is represented as 2's complement. An efficient algorithm to solve this problem is shown in Fig. 2. Calculate 5×-6 step by step with the alogrithm. An example of the following algorithm is shown in Table 1, whre final result 0001,0101 (21_{10}) is stored in A,Q after the last operation.

Table 1: Example of Calculating 7×3

Cycle	Operation	A	Q	Q_{-1}	M
	Initial Value	0000	0011	0	0111
Cycle 1	A←A-M	1001	0011	0	0111
	Shift	1100	1001	1	0111
Cycle 2	Shift	1110	0100	1	0111
Cycle 3	A←A+M	0101	0100	1	0111
	Shift	0010	1010	0	0111
Cycle 4	Shift	0001	0101	0	0111

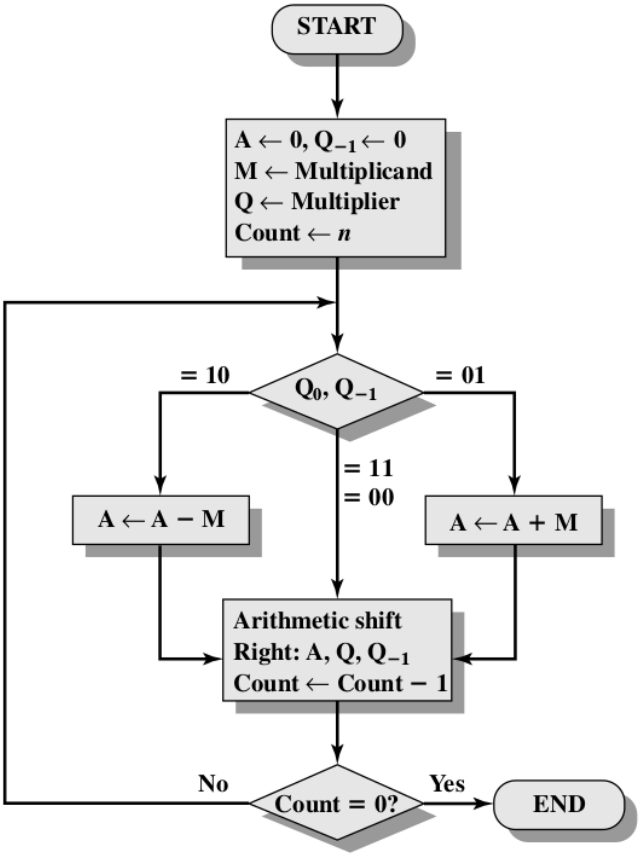

Figure 2: Q3

A3 1. Trivial

2. As shown in Table 2, result is stored in A, Q.

Q4 (**10%**) Consider a single-level cache with an access time of 2.5 ns, a line size of 64 bytes, and a hit ratio of H = 0.95. Main memory uses a block transfer capability that has a firstword (4 bytes) access time of 50 ns and an access time of 5 ns for each word thereafter.

1. What is the access time when there is a cache miss?
2. Suppose that increasing the line size to 128 bytes increases the H to 0.97. Does this reduce the average memory access time?

A4 1. $T_{miss} = 2.5 + 50 + (15)(5) + 2.5 = 130ns$

Table 2: Step by step result of Q3-2

Cycle	Operation	A	Q	Q_{-1}	M
	Initial Value	0000	1010	0	0101
Cycle 1	Shift	0000	0101	0	0101
Cycle 2	A←A-M	1011	0101	0	0101
	Shift	1101	1010	1	0101
Cycle 3	A←A+M	0010	1010	1	0101
	Shift	0001	0101	0	0101
Cycle 4	A←A-M	1100	0101	0	0101
	Shift	1110	0010	1	0101

2. $T_s = (0.95)(2.5) + (0.05)(130) = 8.875ns$
   After modification,
   $T_s = (0.97)(2.5) + (0.03)(210) = 8.725ns$
   Therefore average memory access time is reduced.

Q5 (**10%**) Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used with this machine.

1. How is a 16-bit memory address divided into tag, line number, and byte number?
2. Why is the tag also stored in the cache?

A5
1. 8 leftmost bits = tag; 5 middle bits = line number; 3 rightmost bits = byte number.
2. Because two items with two different memory addresses can be stored in the same place in the cache. The tag is used to distinguish between them.

Q6 (**10%**) Consider a memory system that uses a 32-bit address to address at the byte level, plus a cache that uses a 64-byte line size.

1. Assume a direct mapped cache with a tag field in the address of 20 bits. Show the address format and determine the following parameters: number of addressable units, number of blocks in main memory, number of lines in cache, size of tag.
2. Assume a four-way set-associative cache with a tag field in the address of 9 bits. Show the address format and determine the following parameters: number of addressable units, number of blocks in main memory, number of lines in set, number of sets in cache, number of lines in cache, size of tag.

A6
1. Address format: Tag = 20 bits; Block = 6 bits; Byte addr = 6 bits.
   Number of addressable units = 2^{32} bytes; number of blocks in main memory = 2^{26} ; number of lines in cache = 2^6 = 64; size of tag = 20 bits.
2. Address format: Tag = 9 bits; Set = 17 bits; Byte addr = 6 bits.
   Number of addressable units = 2^{32} bytes; Number of blocks in main memory = 2^{26} ; Number of lines in set = 4; Number of sets in cache = 2^{17} ; Number of lines in cache =2^{19}; Size of tag = 9 bits.

Q7 (**10%**) In this exercise we look at memory locality properties of matrix computation. The following code is written in C, where elements within the same row are stored contiguously.

```
for (J=0; J<8000; J++)
  for (I=0; I<8; I++)
    A[I][J]=B[J][0]+A[J][I];
```

1. How many 32-bit integers can be stored in a 16-byte cache line?
2. References to which variables exhibit temporal locality?
3. References to which variables exhibit spatial locality?

A7 1. 4

2. I, J, B[J][0]

3. A[J][I]

Q8 (**15%**) Consider two processors **a** & **b**. Each processor has five stages with latencies shown in the following table. Assume that when pipelining, each pipeline stage costs **20ps** extra for the registers between pipeline stages.

	IF	ID	EX	MEM	WB
processor a	300ps	400ps	350ps	550ps	100ps
processor b	200ps	150ps	100ps	190ps	140ps

1. If **no** pipeline is considered, for each processor please calculate the following metrics: cycle time; latency of an instruction; the throughput. (Note that throughput is instruction number per second)
2. If both a & b are pipelined processors, for each one calculate the following metrics: cycle time; latency of an instruction; the throughput.
3. If you could split one of the pipeline stages into 2 equal halves, which one would you choose for processors a & b, respectively? For the two processors: what is the new cycle time, the new latency of an instruction, and the new throughput?

A8 1.

- Cycle time: Since no pipeline is applied, cycle time is the summation of the latency of each stage.

$$CT_a = IF_a + ID_a + EX_a + MEM_a + WB_a = 1700ps,$$
$$CT_b = IF_b + ID_b + EX_b + MEM_b + WB_b = 780ps.$$

- Instruction Latency: The same as cycle time.
- Throughput: $Thp = \frac{1}{CT}$.

$$Thp_a = \frac{1}{1700} \times 10^{12} = 5.9 \times 10^8,$$
$$Thp_b = \frac{1}{780} \times 10^{12} = 1.28 \times 10^9.$$

2.

- Cycle time in pipelined processor is determined by the longest stage plus the register latency. Therefore, $CT_a = 570ps$ and $CT_b = 220ps$.
- The latency of processor a is $2850ps$ and the latency of processor b is $1100ps$.
-

$$Thp_a = \frac{1}{570} \times 10^{12} = 1.75 \times 10^9,$$
$$Thp_b = \frac{1}{220} \times 10^{12} = 4.55 \times 10^9.$$

3.

- The cycle time can be reduced by splitting the longest stage i.e. MEM in processor a and IF in processor b.
- $CT_a = 420ps$ and $CT_b = 210ps$.
- The latency of processor a is $2520ps$ and the latency of processor b is $1260ps$.
- $Thp_a = 2.38 \times 10^9$ and $Thp_b = 4.76 \times 10^9$.