Design of Digital Circuits (252-0028-00L), Spring 2018

Optional HW 6: Systolic Arrays, Caches, and Virtual Memory

SOLUTIONS

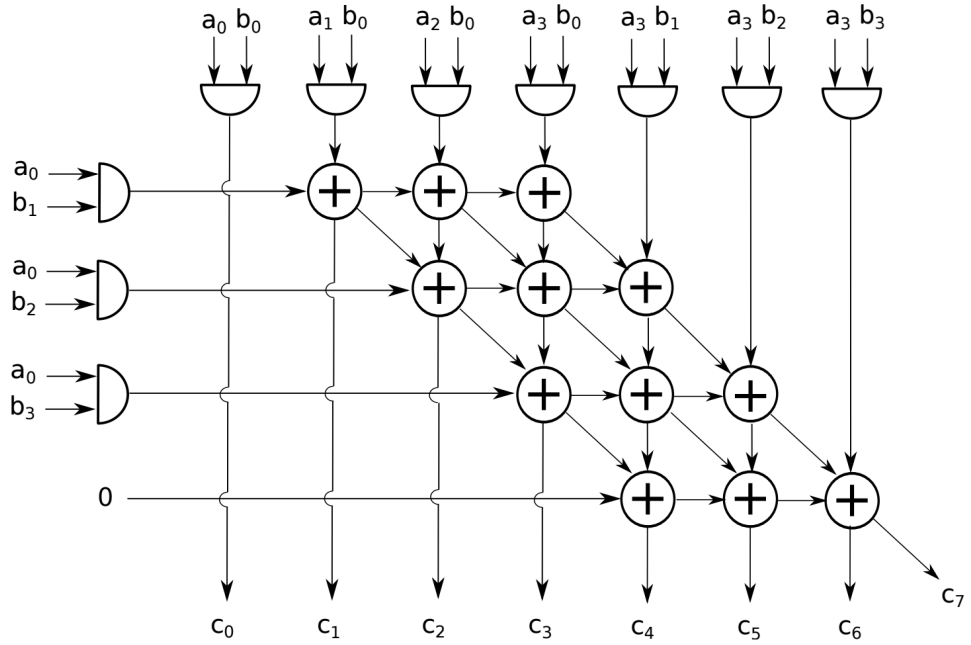
Instructor: Prof. Onur Mutlu

TAs: Juan Gomez Luna, Hasan Hassan, Arash Tavakkol, Minesh Patel, Jeremie Kim, Giray Yaglikci

Assigned: Friday, June 8, 2018

1 Systolic Arrays

The following diagram is a systolic array that performs the multiplication of two 4-bit binary numbers. Assume that each adder takes one cycle.



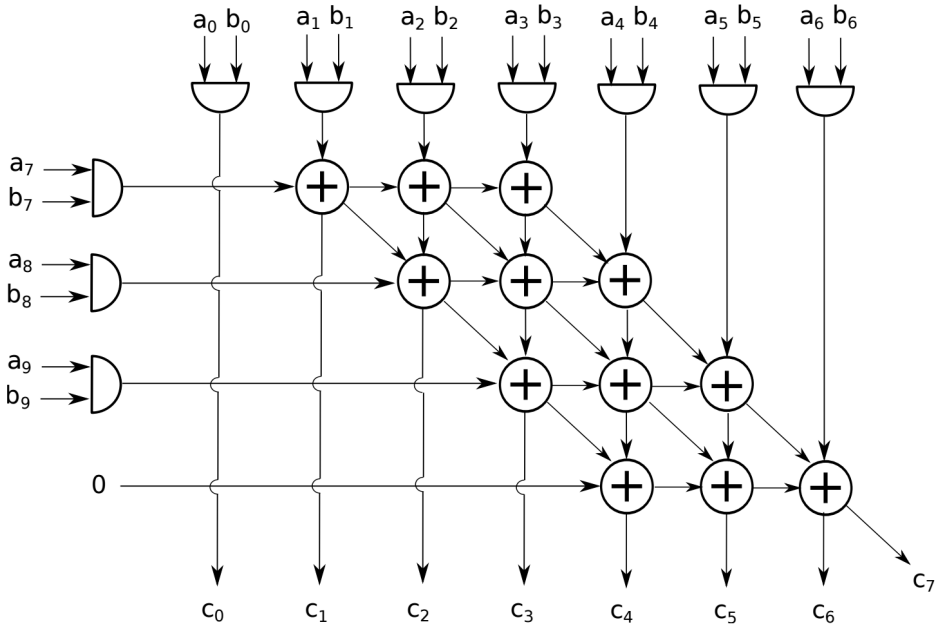
(a) How many cycles does it take to perform *one* multiplication?

9 cycles

(b) How many cycles does it take to perform *three* multiplications?

13 cycles.
Note that some input ports have to hold their values for an extra cycle because their nodes need to wait for inputs from the previous nodes. You cannot completely overlap latencies of different nodes. However, because no one was aware of this, we accepted 11 cycles as another possible answer.

(c) Fill in the following table, which has a list of inputs (a 1-bit binary number) such that the systolic array produces the following outputs, in order: $5 * 5, 12 * 9, 11 * 15$. Please refer to the following diagram to use as reference for all the input ports.



$5x5 = 101x101$ $12x9 = 1100x1001$ $11x15 = 1011x1111$

Cycles	Row Inputs														Column Inputs					
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	b_0	b_1	b_2	b_3	b_4	b_5	b_6	a_7	a_8	a_9	b_7	b_8	b_9
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
2	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0
4	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	1	0
5	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0
6	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	1	1	0	0
7	0	0	0	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	1	1
8	0	0	0	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	1	1
9	1	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0
10	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1
11	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	0	1	0	0	1
12	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
13	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2 Instruction and Data Caches

Consider the following loop is executed on a system with a small instruction cache (I-cache) of size 16 B. The data cache (D-cache) is fully associative of size 1 KB. Both caches use 16-byte blocks. The instruction length is 4 B. The initial value of register `$1` is 40. The value of `$0` is 0.

```
Loop: lw   $6, X($1)
      addi $6, $6, 1
      sw   $6, Y($1)
      subi $1, $1, 4
      beq $1, $0, Exit
      j    Loop
Exit:  ...
```

(a) Compute I-cache and D-cache miss rates, considering:

- `X` and `Y` are different arrays.
- `X` and `Y` are the same array.

The I-cache can keep 4 instructions. Thus, in each iteration there will be 2 cache misses. The I-cache miss rate will be $2/6 = 0.33$.

- `X` and `Y` are different arrays.
  If `X` and `Y` are different arrays, there will be 2 cache misses every 4 iterations, that is, one read miss and one write miss every 8 accesses. In that case, the D-cache miss rate will be $2/8 = 0.25$.
- `X` and `Y` are the same array.
  If `X` and `Y` are the same array, the D-cache miss rate will be one half of the previous one (0.125).

(b) Compute the average number of cycles per instruction (CPI), using a baseline ideal CPI (ideal caches) equal to 2, and a miss latency equal to 10 clock cycles.

Since load and store instructions are one third of all the execute instructions, CPI is calculated as:
CPI = 2 + 0.33×10 + 0.33×0.25×10 = 6.1250 cycles, if `X` and `Y` are different arrays.
CPI = 2 + 0.33×10 + 0.33×0.125×10 = 5.7125 cycles, if `X` and `Y` are the same array.

(c) A compiler could unroll this loop for optimization. How would this affect CPI?

If the compiler unrolls the loop, the total number of executed instructions is reduced. As there will be one I-cache miss every four executed instructions, the I-cache miss rate will be $1/4 = 0.25$. The D-cache miss rate remains the same, but the fraction of loads and stores changes. `beq` and `j` instructions are no longer necessary, so the fraction of load and stores is 0.50.
The new CPI is:
CPI = 2 + 0.25×10 + 0.50×0.25×10 = 5.75 cycles, if `X` and `Y` are different arrays.
CPI = 2 + 0.25×10 + 0.50×0.125×10 = 5.125 cycles, if `X` and `Y` are the same array.

(d) How would the previous results change with a 32-byte I-cache?

If the size of the I-cache is 32 B, the entire loop fits in it. There will be only two cold misses in the first iteration. Thus, the I-cache miss rate will be $2/60 = 0.033$.

3 Reverse Engineering Caches

You're trying to reverse-engineer the characteristics of a cache in a system so that you can design a more efficient, machine-specific implementation of an algorithm you're working on. To do so, you've come up with four patterns that access various *bytes* in the system in an attempt to determine the following four cache characteristics:

- Cache block size (8, 16, 32, 64, or 128 B)
- Cache associativity (1-, 2-, 4-, or 8-way)
- Cache size (4 or 8 KB)
- Cache replacement policy (LRU or FIFO)

However, the only statistic that you can collect on this system is cache hit rate after performing the access pattern. Here is what you observe:

Access Pattern	Blocks Accessed (Oldest → Youngest)									Hit Rate
A	0	4096	8192	12288	16384	4096	0			1/7
B	0	1024	2048	3072	4096	5120	6144	3072	0	1/9
C	0	4	8	16	32	64	128	256	512	4/9
D	128	1152	2176	3200	128	4224	1152			2/7

Based on what you observe, what are the following characteristics of the cache? (Be sure to justify clearly your answer for full credit.)

(a) Cache block size (8, 16, 32, 64, or 128 B)?

Let's focus on access pattern C for this part. In this access pattern, for a given cache block size, all bytes map to certain sets—regardless of cache associativity and regardless of cache size.
There is only one mapping that causes 4 accesses out of 9 to be hits: At a cache block size of 64 B, addresses 4, 8, 16, and 32 all hit in the block brought in by the access to address 0. The other accesses go to different cache blocks.

(b) Cache associativity (1-, 2-, 4-, or 8-way)?

Let's focus on access pattern A for this part. In this access pattern, for a given cache associativity, all bytes map to the *same* set—regardless of cache block size and regardless of cache size.
There is only one associativity that causes 1 access out of 7 to be a hit: An associativity of 4 allows address 4096 to be a hit. Any less, and no accesses would hit; any more, and address 0 would also hit.

(c) Cache size (4 or 8 KB)?

Let's focus on access pattern B for this part. In this access pattern, given a cache associativity of 4, and for a given cache size, all bytes map to the *same* set—regardless of cache block size.
There is only one cache size that maps blocks to sets in a way that causes 1 access out of 9 to be a hit: A cache size of 4 KB causes all of the blocks to map to the same set, generating a hit for address 3072. At 8 KB, the access to address 0 also hits in the cache.

(d) Cache replacement policy (LRU or FIFO)?

Let's focus on access pattern D for this part. In this access pattern, given a cache associativity of 4, and given a cache size of 4 KB, for a given replacement policy, all bytes map to the same set—regardless of cache block size.
First, notice that the access to address 128 will be a hit: Under LRU, the next block to be evicted would be the one for address 1152; whereas, under FIFO, the next block to be evicted would be the one for address 128.
Second, notice that the access to address 4224 then evicts the respective block. The access for address 1152 would then miss under LRU, but hit under FIFO, for a total of two hits (including the access to address 128). FIFO is the only cache replacement policy that causes 2 accesses out of 7 to be hits.

4 Analyzing Cache Structure

Below, we have given you four different sequences of addresses generated by a program running on a processor with a data cache. Cache hit ratio for each sequence is also shown below. Assuming that the cache is initially empty at the beginning of each sequence, find out the following parameters of the processor's data cache:

- Associativity (1, 2 or 4 ways)
- Block size (1, 2, 4, 8, 16, or 32 bytes)
- Total cache size (256 B, or 512 B)
- Replacement policy (LRU or FIFO)

Assumptions: all memory accesses are one byte accesses. All addresses are byte addresses.

Sequence No.	Address Sequence	Hit Ratio
1	0, 2, 4, 8, 16, 32	0.33
2	0, 512, 1024, 1536, 2048, 1536, 1024, 512, 0	0.33
3	0, 64, 128, 256, 512, 256, 128, 64, 0	0.33
4	0, 512, 1024, 0, 1536, 0, 2048, 512	0.25

Cache block size - 8 bytes
For sequence 1, only 2 out of the 6 accesses (specifically those to addresses 2 and 4) can hit in the cache, as the hit ratio is 0.33. With any other cache block size but 8 bytes, the hit ratio is either smaller or larger than 0.33. Therefore, the cache block size is 8 bytes.

Associativity - 4
For sequence 2, blocks 0, 512, 1024 and 1536 are the only ones that are reused and could potentially result in cache hits when they are accessed the second time. Three of these four blocks should hit in the cache when accessed for the second time to give a hit rate of 0.33 (3/9).
Given that the block size is 8 and for either cache size (256B or 512B), all of these blocks map to set 0. Hence, an associativity of 1 or 2 would cause at most one or two of these four blocks to be present in the cache when they are accessed for the second time, resulting in a maximum possible hit rate of less than 3/9. However, the hit rate for this sequence is 3/9. Therefore, an associativity of 4 is the only one that could potentially give a hit rate of 0.33 (3/9).

Total cache size - 256 B
For sequence 3, a total cache size of 512 B will give a hit rate of 4/9 with a 4-way associative cache and 8 byte blocks regardless of the replacement policy, which is higher than 0.33. Therefore, the total cache size is 256 bytes.

Replacement policy - LRU
For the aforementioned cache parameters, all cache lines in sequence 4 map to set 0. If a FIFO replacement policy were used, the hit ratio would be 3/8, whereas if an LRU replacement policy were used, the hit ratio would be 1/4. Therefore, the replacement policy is LRU.

5 Caches

A byte-addressable system with 16-bit addresses ships with a two-way set associative, writeback cache with perfect LRU replacement. The tag store (including the tag and all other meta-data) requires a total of 4352 bits of storage. What is the block size of the cache? Assume that the LRU information is maintained on a per-set basis as a single bit. (Hint: $4352 = 2^{12} + 2^8$.)

We formulate two basic equations:

$$4352 = 2^{index} * (2 * (tag + dirty + valid) + LRU) \quad (1)$$

$$tag + index + offset = 16 \quad (2)$$

There is one dirty bit and one valid bit per block, and one LRU bit per set. So, now the Equation 1 looks like: $4352 = 2^{index} * (2 * (tag + 2) + 1) = (2^{index} * (2tag + 4)) + 2^{index}$

By using the hint ($4352 = 2^{12} + 2^8$), we get $2^{index} = 2^8$, so $index = 8$, and from $2^{index} * (2tag + 4) = 2^{12}$ we get $(2tag + 4) = 2^4$, so $tag = 6$.

By solving the Equation 2, we get $offset = 2$, so, the block size is $2^2 =$ **4 bytes**

6 Virtual Memory

An ISA supports an 8-bit, byte-addressable virtual address space. The corresponding physical memory has only 128 bytes. Each page contains 16 bytes. A simple, one-level translation scheme is used and the page table resides in physical memory. The initial contents of the frames of physical memory are shown below.

Frame Number	Frame Contents
0	Empty
1	Page 13
2	Page 5
3	Page 2
4	Empty
5	Page 0
6	Empty
7	Page Table

A three-entry translation lookaside buffer that uses Least Recently-Used (LRU) replacement is added to this system. Initially, this TLB contains the entries for pages 0, 2, and 13. For the following sequence of references, put a circle around those that generate a TLB hit and put a rectangle around those that generate a page fault. What is the hit rate of the TLB for this sequence of references? (Note: LRU policy is used to select pages for replacement in physical memory.)

References (to pages): 0, 13, 5, 2, 14, 14, 13, 6, 6, 13, 15, 14, 15, 13, 4, 3.

References (to pages): (0), (13), 5, 2, [14], (14), 13, [6], (6), (13), [15], 14, (15), (13), [4], [3].
TLB Hit Rate = 7/16

(a) At the end of this sequence, what three entries are contained in the TLB?

4, 13, 3

(b) What are the contents of the 8 physical frames?

Pages 14, 13, 3, 2, 6, 4, 15, Page table