

ECE 4750 Computer Architecture

Section 10: Integrating Processors and Memories

<http://www.csl.cornell.edu/courses/ece4750>
School of Electrical and Computer Engineering
Cornell University

revision: 2022-11-03-13-43

List of Problems

1	Evaluating a Dot Product Microbenchmark	2
1.A	Analyzing the Average Memory Access Latency	3
1.B	TinyRV1 Single-Issue Scalar Processor	4
1.C	TinyRV1 Dual-Issue Superscalar Processor	5
1.D	Two-Way Set Associative Cache	6

Problem 1. Evaluating a Dot Product Microbenchmark

In this problem, we will explore a dot product microbenchmark executing on a single-issue scalar processor and a dual-issue superscalar processor integrated with either a direct-mapped or set-associative data cache. Here is the C code for the microbenchmark:

```
int dot( int* a, int* b, int n )
{
    int result = 0;
    for ( int i = 0; i < n; i++ )
        result += a[i] * b[i];
    return result;
}
```

And here is the corresponding assembly:

```
addi x10, 0, 0

loop:
lw   x5, 0(x11)
lw   x6, 0(x12)
addi x11, x11, 4
addi x12, x12, 4
mul  x7, x5, x6
addi x13, 1, -1
add  x10, x10, x7
bne  x13, x0, loop

jr   x1
```

Make sure you understand the connection between the C program and assembly before continuing.

For this problem, you should assume a fully bypassed processor that implements the TinyRV1 instruction set. You should assume there an instruction cache with a single-cycle hit latency and a 100% hit rate. You should assume a 256B data cache with 16B cache lines, parallel-read/pipelined-write, a write-back/write-allocate write policy, and a miss penalty of two cycles. Assume the data cache is initially empty.

Assume that we call the dot function with two arrays each with 64 elements (i.e., n is 64). Assume the base address of array a is 0x1000 and the base address of array b is 0x2000.

