Universidad Carlos III de Madrid

Department of informatics
Computer Science degree
Computer Architecture

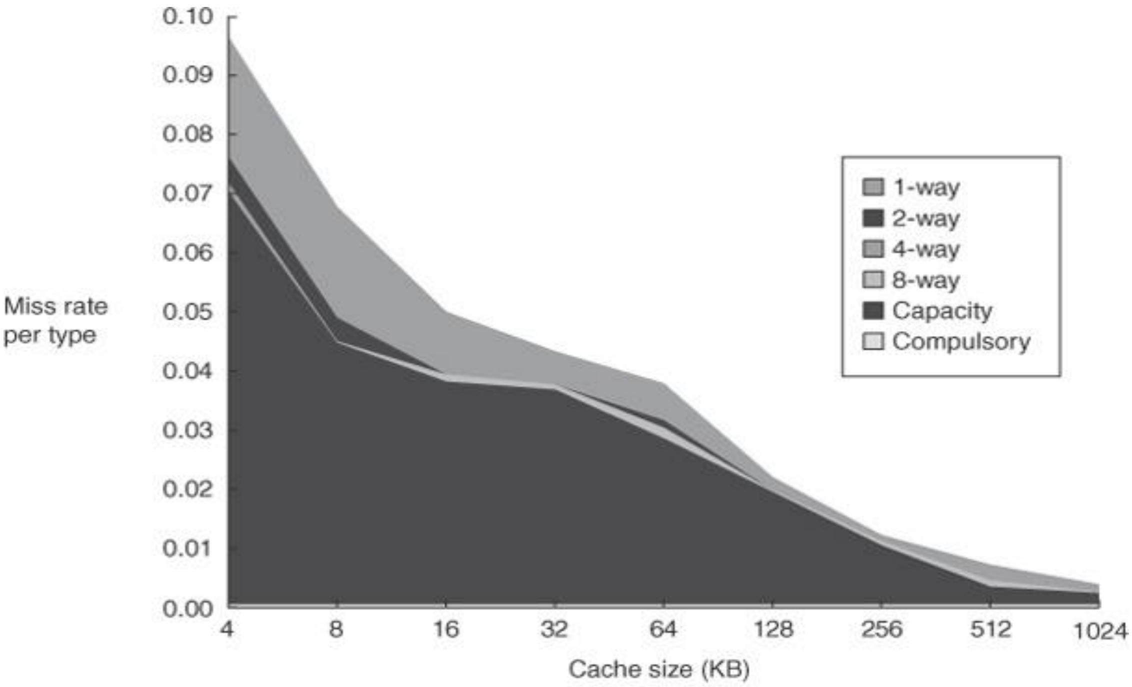
Extra exercises

EXERCISE 1.

Answer the following questions:

- Given a four core processor at 1GHz. We have an application which is 80% parallel. The sequential execution time is 1000 seconds. Explain which alternative will have a smaller execution time:
  a. Run the code sequentially in one core with overclocking at a frequency of 2 GHz.
  b. Run the parallel core in the four cores at the original frequency of 1 GHz.

- Given the following figure, (1) define the capacity and compulsory cache misses; (2) explain the behavior of the cache misses under different n-way set associative configuration. (3) what conclusion is obtained from this figure?



EXERCISE 2.

Given the following code:

```
ADD        R1,R1,1
ADD        R2,R2,1
LD         R4,0(16)
MULT  R5,R4,R4
ADD        R3,R3,R3
SUB        R4,R4,R4
SD         R5,0(16)
DIV        R1,R3,R1
```

There is a MIPS processor with the following phases: *fetch*, decode, execution, memory and write-back. **There is not forwarding and instructions cannot be reordered.**

You are asked to:

a) Indicate all data dependencies.
b) Show in the table from below how the first program iteration proceeds through the MIPS processor.

	Execution cycles															
Instruction	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
ADD																
ADD																
LD																
MULTD																
ADD																
SUB																
SD																
DIV																

c) How many cycles require the whole program to execute?

d) Show in the table from below how the first program iteration proceeds through the **2-way pipeline**. Assume that the pipeline is completely replicated (2-way superscalar) thus the processor issues two instructions per cycle, the decode unit is able to decode to instructions per cycle, there are two ALUs, the memory is able to perform two accesses per cycle and register file can perform four accesses per cycle. The column labeled way shows the pipeline's way where the instruction is being executed. Note that it is not possible to change the way after an instruction is issued.

		Execution cycles															
Instruction	**Way**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
ADD																	
ADD																	
LD																	
MULTD																	
ADD																	
SUB																	
SD																	
DIV																	

EXERCISE 3:

Given a single core computer used for running a financial application. This application uses 90% of the execution time for computational intensive operations. The remaining 10% is spent waiting for Input/Output hard disk accesses. The computational intensive part is divided in 75% for floating point operations and 25% for other instructions. The average CPI of a floating point operation is 12. The rest of the instructions have an average CPI of 4.

We have two different alternative platforms for running this application:

- **Alternative A**: A single core processor with a clock frequency 50% higher than the original computer. For this new system the floating point instructions take 10% more clock cycles per instruction and the rest of the instructions take 25% more cycles than the original ones. The Input/Output time is the same as the original platform.
- **Alternative B**: A four core processor with a frequency 50% smaller than the original computer. The floating point instructions require 20% less clock cycles than the original ones. The rest of the instructions take the same amount of clock cycles as the original ones. The Input/Output time is the same as the original platform.

Answer the following questions:

a) What would be the overall speedup of Alternative A?

b) What would be the overall speedup of Alternative B? Assume that the computational intensive part is completely parallelizable and that the Input/Output part is sequential.

EXERCISE 4

A given processor is intended to run the following code fragment:

```
i0: lw $r4, 0($r1)
i1: lw $r5, 0($r2)
i2: add $r4, $r4, $r5
i3: sw $r4, 0($r3)
i4: addi $r1, $r1, 4
i5: addi $r2, $r2, 4
i6: addi $r3, $r3, 4
i7: bne $r3, $r0, i0
```

Assume that the processor has a segmented architecture of 5 steps (fetch, decode, execute, memory and post-writing) without forwarding. All operations are executed in one cycle per stage, except:

- Load and store instructions, that require two cycles for the memory stage (an additional cycle).
- Branch instructions require an additional cycle in the execution stage. Assume that these instructions do not have any branch prediction.

Answer the following questions:

a) Obtain the RAW risks.
b) Show the timing diagram with the phases of execution of each instruction for one iteration.
c) Determine how many cycles it takes to execute a single iteration if there is no branch prediction.
d) Propose a loop unrolling solution, assuming that the loop runs 1000 iterations. Unroll with a factor of four iterations and write the resulting code.
e) Determine the speedup obtained with loop unrolling.

EXERCISE 5:

A given processor has associated the following latency table:

Instruction producing result	Instruction using result	Latency
ALU FP operation	Other ALU FP operation	6
ALU FP operation	Store double	3
Load double	ALU FP operation	2
Load double	Store double	0

The following code is executed in this processor:

```
LOOP:
L.D F0, 0(R1)
L.D F2, 0(R2)
ADD.D F4, F0, F2
S.D F4, 0(R3)
DADDUI R1, R1, #-8
BNE R1, R4, LOOP
```

The initial values of the registers are the following ones:

- R1: Last element of the first source array
- R2: Last element of the second source array
- R3: Last element of the destination array
- R4: Predefined: 8(R4) is the first element of the first source array.

All the arrays have 4000 entries.

Complete the following tasks:

1. Obtain the number of cycles necessary to execute all the iterations of the original code.
2. Obtain the number of cycles necessary to execute all the iterations of the code when you rewrite code to minimize stalls.
3. Obtain the number of cycles necessary to execute all the iterations of the code when you unroll the loop two times.
4. Obtain the number of cycles necessary to execute all the iterations of the code when you unroll the loop four times.

EXERCISE 6

The following program is run on a machine with a cache that has four-word (16 byte) blocks and holds 256 bytes of data. Assume that:

- only the cache activity generated by references to the array
- the integers are words.
- The memory address of array[0] is 0

Answer the following questions:

a) What is the expected miss rate when the cache is direct-mapped and stride = 132? Which types of misses are encountered?

b) How about if the stride = 131? Which types of misses are encountered?

c) Would either of these change if the cache were two-way set-associative? Which types of misses are encountered?

```
int i, j, c, stride, array[256];
……
```

```
for (i = 0; i < 10000; i++)
    for (j = 0; j < 256; j = j + stride)
        c = array[j] + 5;
```

EXERCISE 7

On a 32 bits computer having a CPU with the frequency 1GHz (cycle of 1ns) the following program executes 6144 instructions and accesses 2048 data elements through memory load instructions.

```
I1:        LD         F1,0(R1)
I2:        MULTD F3,F1,F1
I3:        ADDI       R1,R2,#4
I4:        LD         F1,0(R1)
I5:        MULTD F3,F1,F1
I6:        ADDI       R1,R2,#4
                      …
I6142:     LD         F1,0(R1)
I6143:     MULTD F3,F1,F1
I6144:     ADDI       R1,R2,#4
```

Note that the program does not have neither store or jump instructions and that the accessed memeory locations are consecutive. The memory hierarchy has the following characteristics:

- Each instructions and data element occupies one word. The cache line is 16 bytes long.
- The level 1 (L1) instruction cache has 4KB, has an access time of 1 cycle, is directly mapped and is initially empty. When the cache is full and hardware prefetching is used the miss rate is 10% .
- The level 2 (L2) instruction cache has 4KB, has an access time of 2 cycles, is directly mapped and is initially empty. When the cache is full and hardware prefetching is used the miss rate is 5%.
- The level 2 (L2) cache is used for both instructions and data, is totally associative, has an access time of 5 cycles, has a miss rate of 1% and is initially full
- The main memory has a an access time of 15 cyckes.

You are asked to:

1. Calculate the miss rates of caches L1 and L2 assuming that they are initially empty.
2. Calculate and justify the average memory access time.
3. Explain how the results computed in this exercise would change if using a four bank memory instead on one bank memory.

EXERCISE 8

Given the following code:

```
   DADDUI R3, R1, #40
loop:
   L.D F0, 0(R1)
   L.D F2, 0(R2)
   ADD.D F4, F0, F2
   S.D F4, 0(R1)
   DADDUI R1, R1, #8
   DADDUI R2, R2, #8
   BLE R1, R3, loop
```

and the following latencies between instuctions:

Instruction producing result (previous)	Instruction using result (next)	Latency (clock cycles)
FP ALU op	FP ALU op	5
FP ALU op	Store/load double	4
Load double	FP ALU op	2
Load double	Load double	1
Store double	FP ALU op	2
FP ALU op	Jump	4

and assuming that a jump has a latency of 1 cycle and does not have any delay slot, answer the following questions:

a) Identify data dependencies.

b) Assume that: you have one execution pipeline, each executing one instruction per cycle, you have enough fetch/decode bandwidth and no instruction reordering is performed. How many cycles does the loop require?

c) Reorder the instructions to improve performance. How many cycles does the loop require? What is the speedup over the code without reordering?

d) Modify the code by unrolling two loop iterations. What is the speedup over the code without reordering?

EXERCISE 9

The following MIPS code reads and writes the same memory address several times. Assume that initially R1=1 and R2=1000.

```
            ADDI R3,R3,1
Loop:       LD   R4,0(16)
            MULTD R5,R4,R4
            ADDD R5,R3,R5
            SD   R5,0(16)
            DADDI R3,R4,1
            DADDI R1,R1,1
            BNE  R1,R2,Loop
```

Assume that the code executes on a MIPS processor having the following phases: *fetch*, decode, execution, memory and write-back. The processor issues one instruction per cycle and has *forwarding.* All instructions require one cycle to execute except SD, LD, ADDD and MULTD, which require two cycles. Assume that the branch prediction strategy is ***not-taken***.

You are asked to:

e) Indicate all data dependencies.
f) Indicate the instructions that cannot be reordered.
g) Show in the table from below how the first program iteration proceeds through the pipeline assuming *forwarding.*
h) How many cycles requires the ***whole*** program to execute?
i) How many cycles requires the ***whole*** program to execute if the branch prediction strategy is changed to **taken**?

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
ADDI	F	D	E	M	W														
LD																			
MULTD																			
ADDD																			
SD																			
DADDI																			
DADDI																			
BNE																			

EXERCISE 10

Given the following code and the total number of clock cycles employed per instruction:

```
Loop:  LD     F2,0(Rx)
 I0:   DIVD   F8,F2,F0
 I1:   MULTD  F2,F6,F2
 I2:   LD     F4,0(Ry)
 I3:   ADDD   F4,F0,F4
 I4:   ADDD   F10,F8,F2
 I5:   ADDI   Rx,Rx,#8
 I6:   ADDI   Ry,Ry,#8
 I7:   SD     F4,0(Ry)
 I8:   SUB    R20,R4,Rx
 I9:   BNZ    R20,Loop
```

Execution time	
Memory LD:	4
Memory SD:	1
ADDI, SUB	1
Branches	2
ADDD	3
MULTD	5
DIVD	9

Answer the following questions:

(a) Assuming we are using a two pipeline architecture where both pipelines can execute any kind of operation at the same time (always fulfiling dependency issues), write the program using loop unrolling and code reordering for a number of iterations multiple of 2.

(b) Compare the speedup obtained in an execution with four iterations using your version of the code and the original one.

EXERCISE 11

Given a cache with a block size of 4 bytes, a total capacity of 16 bytes, a write-allocate policy and a LRU cache replacement policy. Assume that an integer is stored on 4 bytes, and an array is stored in memory in row order. For the following program:

```
int i;
int a[5];
for (i=0;i<=3;i++)
   a[i+2]=a[i]+1:
```

both i and a are to be cached and the data is laid out in memory in the following way:

Memory address	Variable
0	i
4	a[0]
8	a[1]
12	a[2]
16	a[3]
20	a[4]

I. Identify the types of memory accesses (read or write) in the following table. Note that some instructions result in 2 memory access. The first two lines are already filled in.

II. Identify if the access results in hit or a miss and indicate the type of cache misses (compulsory, capacity, or conflict miss) of the following program, for each of the following three cases: (a) a fully asociative cache (b) a 2-way associative cache (c) direct mapped cache.

III. Calculate the miss rate for each of the three caches.

EXERCISE 12

For the following code:

```
1: MOV R3, R7
2: LD R8, (R3)
3: ADD R3, R3, 4
4: LOAD R9, (R3)
5: BNE R8, R9, L3
```

a) Identify WAW, RAW, and WAR dependencies.

b) What is the difference between a dependency and hazard?

c) What is the difference between a name dependency and a true dependency?

d) Which of WAW, RAW, WAR are true dependencies and which are name dependencies?

EXERCISE 13

Given the following code:

```
1:    Loop:     LD F2,0(Rx)
2:              MULTD F2,F0,F2
3:              DIVD F8,F2,F0
4.              LD F4,0(Ry)
5:              ADDD F4,F0,F4
6:              ADDD F10,F8,F2
7:              SD F4,0(Ry)
8:              ADDI Rx,Rx,#8
9:              ADDI Ry,Ry,#8
10:             SUB R20,R4,Rx
11:             BNZ R20,Loop
```

and the following latencies for the instructions:

Instruction	LD	SD	Integer ADD, SUB	Branches	ADDD	MULTD	DIVD
Latency	4	2	1	2	3	5	11

Answer the following questions:

a) Assuming that: one instruction per cycle is issued, one pipeline is available, the execution is in-order and the branch is taken. How many cycles per iterations are needed to execute one loop? Ignore in your calculations the initial fetch and decode.

b) Identify the true data dependencies in the code.

c) Assume that: you have two execution pipelines, each executing one instruction per cycle, you have enough fetch/decode bandwidth and no instruction reordering is performed. How many cycles does the loop require?

d) Reorder the instructions to improve performance (data dependencies and functional unit latencies must be observed. How many cycles does the loop require? What is the speedup over one pipeline and two pipelines without reordering?

EXERCISE 14

Given a ***fully associative*** 128-byte instruction cache with a 4-byte block (every block can hold exactly one instruction). The cache uses an LRU replacement policy.

a) What is the instruction miss rate for a loop with a very large number of iteration if the loop size is (i) 64-byte (ii) 192 bytes (iii) 320 bytes

b) If the replacement policy is changed to MRU (most recently used), which of the three cases from above would benefit?

c) For the cases in which the miss if high, propose an optimization of the loop that would improve the performance.

EXERCISE 15

Increasing a cache's associativity (with all other parameters kept constant) statistically reduces the miss rate. Given a cache with a total size of 4 bytes, having 4 entries, you are asked to.

a) Construct a sequence of memory accesses (addresses) which provides a worse miss rate for a 2-way associative cache than a direct-mapped cache. Which is the hit rate? (The sequence must be in the form 4,7,8,3, 2 ,4 …, where the numbers represent memory addresses.) Assume the set-associative cache uses a LRU replacement policy.

b) Construct a sequence of memory accesses (addresses) which provides a worse miss rate for a fully associative cache than a two-way associative cache. Which is the hit rate? Assume both caches use a LRU replacement policy.

EXERCISE 16

In the system we are analyzing the memory has:

- Separate L1 instruction and data caches, HitTime = Processor Cycle Time
- 32KB L1 instruction cache with 2% miss rate, 64B blocks
- 256KB L1 data cache with 5% miss rate, 16B blocks
- 256K L2 unified cache with 64B blocks, local miss rate 20%, Hit Time = 4 cycles,
- Main Memory Access time is 50 cycles for the first 64 bits and subsequent 64 bit chunks are available every 10 cycles.
- Both L1 caches are four-way associative, L2 direct mapped.
- Assume there are no misses to main memory.

(a) What is the Miss Penalty for accesses to L2?

(b) What is the average memory access time for instruction references?

(c) What is the average memory access time for data references?

(d) Assume the only memory reference instructions are loads(25%) and stores(5%). What percentage of total memory references are data references?

(e) What is the Average memory access time?

EXERCISE 17

Loop Unrolling - Given the code

```
$L46:
     addu $2, $2, $3
     l.d $f0, 0($2)
     mul.d $f2, $f0, $f0
     add.d $f4, $f2, $f0
     s.d $f4, 0($4)
     subui $4, $4, #8
     beq $2, $4, $L46
```

(a) Show how to unroll the loop just once.

(b) If your loop would normally execute an odd number of times and your unroll it once, what problems arise and how do you address them.

(c) What in the architecture limits the number of times a loop could be unrolled?

EXERCISE 18

Assume a classic 5-stage pipeline that has been extended to allow multiple cycle operations, with the execution stage of FP multiplies taking 4 cycles, loads and stores taking 1 cycle, FP adds take 2 cycles and integer operations take 1 cycle.

Show how the code below moves through the pipeline. Stop with cycle 17.

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
addu $2,$2,$3	F	D	E	M	W														
l.d $f0,0($2)																			
mul.d $f2,$f2,$f0																			
l.d $f8,1000($2)																			
mul.d $f4,$f4,$f8																			
add.d $f6,$f2,$f4																			
s.d $f6,0($4)																			