Exercises Computation II 5EIB0

Answers

Answer 1

```
                        N_instructions   CPI      T_cycle  T_execution
-------------------------------------------------------------------
Single cycle     10000                   1        2.0 ns   20000 ns
Multi cycle              10000           4        0.4 ns   16000 ns
Pipelined                10000           1        0.4 ns    4000 ns
```

Answer 2

```
a. CPI_ideal = 1/3
b. CPI_branch = CPI_ideal + f_branch * f_wrong * BranchPenalty
              = 0.33 + 0.15 * 0.05 * 19
              = 0.476
```

Answer 3

6 stall cycles.

```
        lw  $t0, 0($t2)
         2 stall cycles
        lw  $t1, 4($t0)
         2 stall cycles
        sub $s5, $t1, $t2
         2 stall cycles
        sw  $s5, 4($t0)
```

Answer 4

An extra (third) read port is needed.
Check the MIPS pipelined data path figures !!

Answer 5

```
CPI = CPI_ideal + f_inst * I_missrate * I_misspenalty
                + F_data * D_missrate * D_misspenalty
    = 2 + 1 * 0.05 * 20 + 0.3 * 0.1 * 20 = 3.6 cycles
Slowdown is T_new / T_old
     = (N_instr_new * CPI_new * T_cycle_new)/ N_instr_old * CPI_old * T_cycle_old
     = CPI_new / CPI_ideal = 3.6 / 2.0 = 1.8
```

(so if the ideal cache program would take 1000 cycles, the real one takes 1800 cycles, or an 80 % slowdown)
Note that N_instr and T_cycle do not change.

Answer 6

```
Tag bits = 32 = index - word_offset - byte_offset = 32 - 8 - 1 -2 = 21 bits
Cache size = 4 * 2^8 * (value bit + tag bits + block bits)
           = 2^10 * (1 + 21 + 64)
           = 86 kbit
```

Answer 7

The data memory access pattern is: 100, 108, 104, 112, 108, 116, 112, 120

```
This mappes to word:                  0,   2,   1,   3,   2,   4,   3,   5
1-word block:                         M    M         M    M    H    M    H    M  -> 25%
2-word block:                    M    M    H    H    H    M    H    H  -> 62.5%
4-word block:                    M    H    H    H    H    M    H    H  -> 75%
```

Note, there are no capacity or conflict misses.
(making the cache smaller and/or the offset of the second load bigger can introduce these misses).

Answer 8

The page size = 16kB, so its addressing requires 14 bits. That means the virtual page number is 40-14 = 26 bits, and the physical page number = 36-14 = 22 bits.
Total table size = 2^26 * (22+4) bit = 1664 Mbit = 208 Mbyte !!

Answer 9

```
Try all interleavings of the 4 instructions of P1 and the 4 from P2.
1) a + b - c = 90
2) a + b  = 120
3) a - c  = 70
```

Answer 10

```
strcpy:
    addi $sp, $sp, -4      # adjust stack for 1 item
    sw   $s1, 0($sp)       # save $s0
    add  $s1, $zero, $zero # i = 0
L1: add  $t1, $s1, $a1     # addr of y[i] in $t1
    lbu  $t2, 0($t1)       # $t2 = y[i]
    add  $t3, $s1, $a0     # addr of x[i] in $t3
    sb   $t2, 0($t3)       # x[i] = y[i]
    beq  $t2, $zero, L2    # exit loop if y[i] == 0
```

```
    addi $s1, $s1, 1       # i = i + 1
    j    L1                # next iteration of loop
L2: lw   $s1, 0($sp)       # restore saved $s0
    addi $sp, $sp, 4       # pop 1 item from stack
    jr   $ra               # and return
```

Answer 11

a. S = Told / Tnew = 1 / (S + P/10) = 1 / (0.10 + 0.9/10) = 5.26
b. S = Told / Tnew = 1 / (S + P/N) = 1 / (S + (1-S)/N) = N / ((N-1)*S +1)
c. S = Told' / Tnew = (S+N*P) / (S + N*P/N) = S + (1-S)N / (S + (1-S)) = S + (1-S)N
   i.e. Gustafson's law

Answer 12

- Assuming it has a single-precision FP multiply-add instruction,
- Single-precision FP multiply-add performance = #MPs * #SP/MP * #FLOPs/instr/SP * #instr/clock * #clocks/sec = 8 * 8 * 2 * 1 * 1.5 G = 192 GFlops / second
- **The peak DDR3 bandwidth** =
  #Partitions * #bytes/transfer * #transfers/clock * #clocks/sec =
  8 * 8 * 2 * 1G = **128 GB/sec**

a. Total DDR3RAM memory size = 8 * 256 MB = 2048 MB
   Modern computers have 32-bit single precision
   So, if we want 3 n*n SP matrices, maximum n is
   3n^2 * 4 <= 2048 * 1024 * 1024
   n_max = **13377 = n**

b. For each element of the result, we need n multiply-adds
   For each row of the result, we need n * n multiply-adds
   For the entire result matrix, we need **n * n * n multiply-adds**
   Thus, 2393 GFlops.
   Per multiply-add we need to load 2 source operands, 4 bytes each.

   Now a discussion is needed about the bottleneck, either processing of memory bandwidth. If no caching, memory is clearly determining execution speed, for optimal caching (using tiling) its the processing.
   a. **Assuming cache**: loading of 2 matrices and storing of 1 to the graphics memory. That is 3 * n^2 = 512 GB of data =>
      t_memory = 512 / 128 = 4 seconds.
      t_processing = 2393 / 192 = 12.46 seconds
      t_total = **16.46 seconds**.
   b. **No cache:** 2393 GFlops require 2393*2*4 Gbytes (note, storing the result can in this case be neglected) =>
      t_memory = 2393*2*4 / 128 = 149.6 seconds
      t_total = 149.6+12.5 = **162.1 seconds**

Answer 13

2D grid/mesh: n^2 nodes, n=4 in picture

Diameter: n =4
Nodal degree: 4 (assuming unidirectional links)
Network Bandwidth: 2*P*B = n^2*B = 2*16*B = 32*B
Bisection Bandwidth: 2n*B = 8*B

n-cube tree: 2^n nodes, n=3 in picture

Diameter: n = 3
Nodal degree: 2*n = 2*3 = 6 (bidirectional links)
Network Bandwidth: N_links*B = 2^n*2n*B = 24*B
Bisection Bandwidth: 2^n *2 / 2 * B=2^n*B = 8*B

Scalability

- **pro mesh:** constant nodal degree (so cheap), easier to layout in 2 dimensions
- **pro cube:** short diameter

Answer 14

a. In shared memory system: using (regular) loads and stores
   In message passing system by sending and receiving messages
b. Yes, the address space can be fully shared, while physically memories can be distributed. It means that loads and stores can address all locations, also the ones in other cores.
c. Pros of shared memory:
   - well-known programming model;
   - large data structures can be shared (and passed by reference)
   - no memory fragmentation losses

   Cons of shared memory:
   - synchronization, coherence and consistency issues have to be addressed

Exercises Computation II 5EIB0

Questions

Question 1

We compare 3 different implementations of the MIPS architecture, the single cycle, the multi-cycle and the (5-stage) pipelined implementation, all implemented in the same technology. All run the same MIPS program. This program contains 20% load, 10% store, 20% control, and 50% other instructions. On the multi-cycle implementation loads take 5 cycles, control instructions 3 cycles, and all others 4 cycles. To make the exercise easier you may ignore all hazards and pipeline register overhead.

Can you fill in the open fields in the next table.

	N_instructions	CPI	T_cycle	T_execution
Single cycle	10000	-	2.0 ns	-
Multi cycle	-	-	-	-
Pipelined	-	-	-	-

Question 2

A Pentium-4 processor is a so-called 3-issue machine. It has 20 pipeline stages. It uses branch prediction, with a correct branch prediction rate of 95%. A correctly predicted branch has no penalty, however a wrongly predicted branch has a penalty of 19 cycles. 15% of the instructions are branches.

a. What is its ideal CPI of a Pentium-4?
b. What is the CPI if we take branches into account, but ignore all other hazards.

Question 3

Given the 5-stage pipelined MIPS processor from the book (requiring 1/2 cycle for register file read and register file write), with all hazard detection included, but **no** forwarding!
We run the following program:

```
lw  $t0, 0($t2)
lw  $t1, 4($t0)
sub $s5, $t1, $t2
sw  $s5, 4($t0)
```

How many stall cycles are generated during the execution of above program.

Question 4
We add the following 'multiply-accumulate' instruction to the 5-stage pipelined MIPS processor:

```
      mulacc  r1, r2, r3      // r1 = r1 + r2*r3
```

Which changes are needed to the register file of the pipelined data path?

Question 5
Given a MIPS processor running at 1 GHz clock frequency. The memory hierarchy consists of separate level-1 instruction and data caches and main memory. Both level-1 caches have single cycle access and a size of 16 kbyte. The I-cache has 5 % miss rate, while the D-cache has a 10 % miss rate.
Main memory is 0.5 Gbyte and has an access time of 20 cycles. We are running a program for which 30 % of the instructions access data memory.
With an ideal memory (i.e. all hits) this MIPS processor would have a CPI = 2.0

How much is the slow down in speed due to non-ideal memory hierarchy (i.e. due to the cache misses)?

Question 6
Given a 4-way set-associative cache with 2 words per block, and 4 bytes per word, and an index of 8 bits. The processor uses 32-bit addresses, and byte-addressable memory.
How big is this cache in total (including tags etc.).

Question 7
Given a MIPS processor with a very small 64-byte level-1 direct mapped date cache.
On this processor the following code is running:

```
      addi $s0, $zero, 3
      addi $s1, $zero, 100
loop: lw   $t1, 0($s1)
      lw   $t1, 8($s1)
      addi $s1, $s1, 4
      addi $s0, $s0, -1
      bne  $s0, $zero, loop
```

What is the data cache hit-rate for this program, assuming that this cache has 1-word (32-bit) blocks, 2-word or 4-word blocks?

Question 8
Consider a virtual memory system with the following properties: 40-bit virtual byte address, 16 kbyte page size, 36-bit physical byte address. What is the total size (in bits) of the page table for each process on this machine, assuming that each entry contains a

valid, protection, dirty and use bit, and that all virtual pages are in use. Tricks to reduce the page table are not used.

Question 9
Given following two processes, P1 and P2, running in parallel. They have access to the shared variable 'a'. Variables b and c are locals.

```
/* Process P1 */                    /* Process P2 */

lw      $t0,a                       lw      $t2,a
lw      $t1,b                       lw      $t3,c
add     $t0,$t0,t1                  sub     $t2,$t2,$t3
sw      $t0,a                       sw      $t2,a
```

Given that variables a, b and c contain initially the values 100, 20, and 30 respectively, what are all possible outcomes of the value of 'a' after executing both processes once?

Question 10
Give the MIPS assembly code for the following string copy function

```
void strcpy (char x[], char y[])
{
  int i;
  i = 0;
  while ((x[i] = y[i]) != 0)   // copy and test byte
     i++;
}
```

Assume: i located in register $s1, arguments x[] and y[] are in registers $a0 and $a1.

Question 11 Performance scaling (weak and strong; Amdahl vs Gustafson)
Assume you have a program with a serial part (S), which does not show any speedup when executed on a parallel processor, and a parallel part (P), which shows ideal speedup on a parallel processor. Assume the S = 10 % of the program, when running on a single core, with a certain data input with fixed size (D). Give the speedup when running this program on a 10-core system.
Give the general formula for Speedup when running this program on an N-core parallel system?
Now we assume so-called weak scaling, i.e. the amount of input data (D) for the parallel part is proportional to the number of cores (N); give again the general formula for speedup.

Question 12. Parallel execution of Matrix-Matrix multiply

We are building a highly parallel system (like a GPU processor) to support matrix-multiplication. The system contains:

- 8 cores, running at 1.5 GHz.

- each core supports 8 single precision floating point multiply-add operations per cycle.
- main memory: 8 memory banks, each 8 bytes wide, with each 256 MB DRAM capacity; the memory system runs at 1 GHz
- the system does not contain data caches.

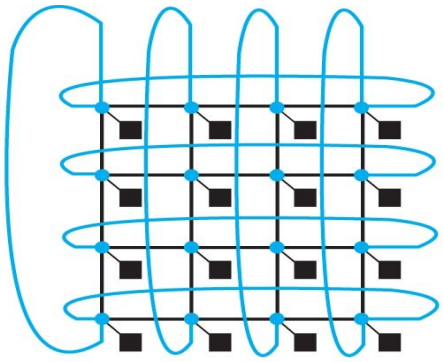
On this system we run matrix-matrix multiplication: C = A*B, where A,B and C are n*n matrices, each element is a single float (of 4 bytes).

a. The maximum size of n is determined by the fitting all 3 matrices in the main memory. What is the maximum size of n?
b. How many seconds will the total matrix multiplication take? Clearly state your assumptions.

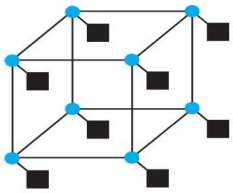
Question 13. Network properties

Besides network bandwidth and bisection bandwidth, two other properties sometimes used to describe network typologies are the diameter and the nodal degree. The diameter of a network is defined as the longest minimal path possible, examining all pairs of nodes. The nodal degree is the number of links connecting to each node.

a. If the bandwidth of each link in a network is B, find the **diameter**, **nodal degree**, **network bandwidth**, and **bisection bandwidth** for the 2D grid of n-by-n nodes and n-cube tree. Examples of these 2 networks are shown below.
b. Which of the 2 networks has better scalability towards larger number of nodes N ($N=n^2$ and n^3 for these 2 networks respectively). Motivate shortly your answer.



a. 2-D grid or mesh of 16 nodes



b. *n*-cube tree of 8 nodes ($8 = 2^3$ so $n = 3$)

Question 14.

Parallel processors can be build in two ways: having a shared memory space (a shared memory system), or having local address spaces per core.

a. How do cores preferably communicate in a shared memory system, and how in the other system?
b. Can a shared memory system have distributed memories, e.g. just one memory per core? Motivate your answer.
c. Give the pros and cons of these 2 systems.

Department of Electrical Engineering, Electronic Systems group

5EIB1– Computation II

Final exam, July 2, 2015, 18.00-21.00 h.

Surname	:	**Initials**	:
Student id	:	**Program (EE, AU, …)**	:

During this exam, the use of computers, calculators or any other material is not allowed. The green card provided with the Patterson& Hennessey book may be used. Provide your answers in the designated areas in this form. Focus on the final answers. The examiner may decide to ignore any additions. *You have to hand in this form before you leave*. The exam consists of 9 questions on 7 pages.

Question 1: MIPS instruction set

a) For the following C statement, what is the corresponding MIPS assembly code? Assume that the variables i, and j are assigned to registers $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively.

```
B[8] = A[i−j];
```

For questions b and c assume that registers $s0 and $s1 hold the values 0x80000000 and 0xD0000000, respectively.

b) What is the value of $t0 for the following assembly code?

```
add $t0, $s0, $s1
```

c) Is the result in $t0 the desired result, or has there been overflow?

d) Give the binary code for the following branch BNE instruction at hex address 0x4004. Clearly indicate all instruction fields.

```
4000  loop:  ......
4004         BNE $1, $5, loop
```

Answer 1

a) sub $t0, $s3, $s4
add $t0, $s6, $t0
lw $t1, 16($t0)
sw $t1, 32($s7)

b) 50000000
c) Overflow
d) loop -> -2d = 1..110, BNE opcode = 000101 => 000101 00001 00101 11111111.11111110

Question 2: MIPS pipeline stalls and code scheduling
Consider the 5-stage pipelined MIPS processor, as described in the book, running the following code (7 instructions):

```
lw   $t0, 64($zero)
lw   $t1, 0($t0)
lw   $t2, 4($t0)
add  $t5, $t5, 8
add  $t3, $t1, $t2
sw   $t3, 12($t0)
lw   $t4, 8($t0)
add  $t5, $t1, $t4
sw   $t5, 16($t0)
```

a) How many cycles are required for the execution of this code when no forwarding circuitry has been implemented? Assume **3 stall cycles** in case of any RaW dependence. Omit the 4 cycles required to fill the pipeline.
b) Same question, but now we add all the possible forwarding (bypassing) circuitry.
c) Give the re-ordered (rescheduled) code for optimal performance, still assuming all the forwarding circuitry.

Answer 2

a) 9+ 14 stalls = 23 cycles

b) 9 + 2 stalls = 11 cycles

c) old schedule; extra latencies without/with forwarding

```
lw   $t0, 64($zero) //
lw   $t1, 0($t0)    // +3/1
lw   $t2, 4($t0)    //
add  $t5, $t5, 8    //
add  $t3, $t1, $t2  // +2
sw   $t3, 12($t0)   // +3
lw   $t4, 8($t0)    //
add  $t5, $t1, $t4  // +3/1
sw   $t5, 16($t0)   // +3
```

New schedule, zero stalls !

```
lw    $t0, 64($zero)
add   $t5, $t5, 8
lw    $t1, 0($t0)    // put first this load, since $t1 is needed
lw    $t4, 8($t0)    // by both adds
add   $t3, $t1, $t2
add   $t5, $t1, $t4
sw    $t3, 12($t0)
sw    $t5, 16($t0)
```

Question 3: MIPS pipelining

We examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies (this includes pipeline stage register delays):

IF	ID	EX	MEM	WP
250 ps	350 ps	150 ps	300 ps	200 ps

Assume that instructions executed by the processor are broken down as follows:

alu	beq	lw	sw
45%	20%	20%	15%

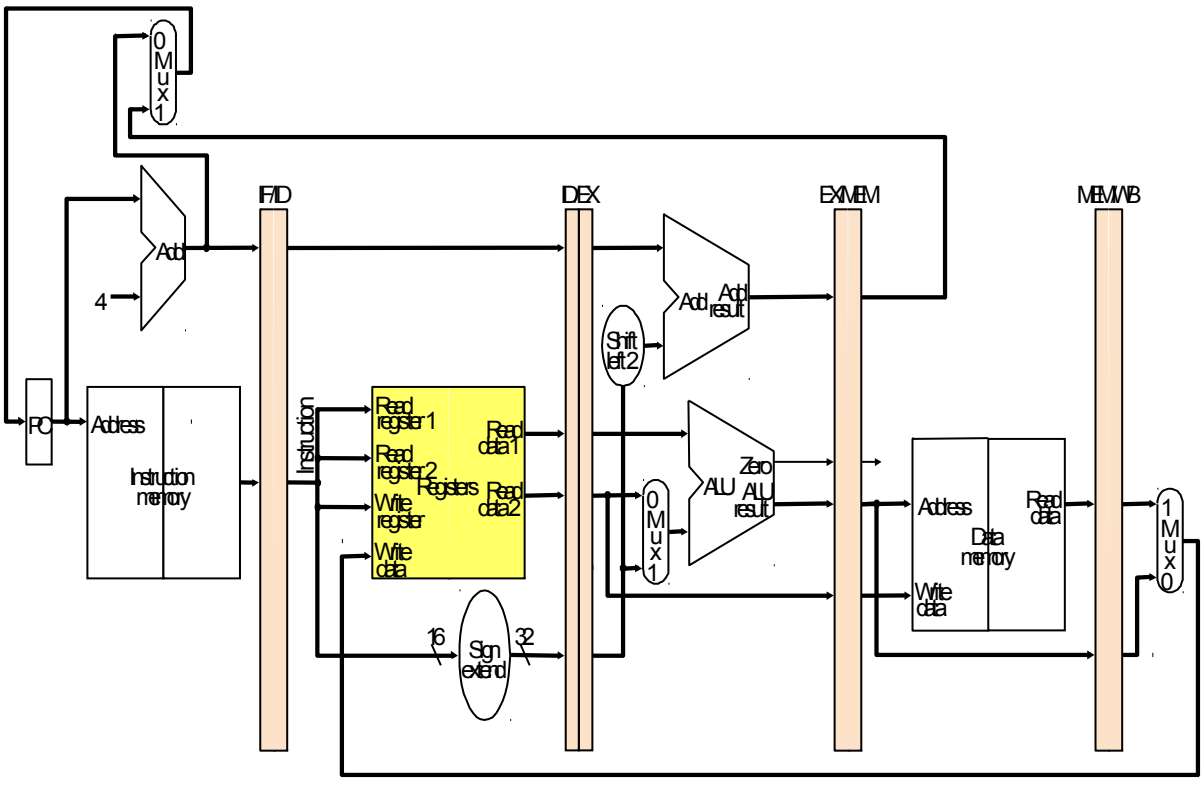
a) What is the clock cycle time in a pipelined and non-pipelined processor?
b) What is the total latency of an LW instruction in a pipelined and non-pipelined processor?
c) Assuming no stalls and hazards (all delay slots are effectively filled), what is the utilization of the write-registor port of the register file?
d) Assume that all branch delay slots are not filled and the delay slots of lw instructions are filled in 50% of the cases. There are no other stalls and hazards. Calculate the average CPI.

Answer

a) 350 ps pipelined, 1250 ps non-pipelined
b) 1750 ps pipelined, 1250 ps non-pipelined
c) 65 % (alu + lw instructions use a write port)
d) CPI = 0.45*1 + 0.2*2 + 0.2*1.5 + 0.15 = 1.4 cycles/instruction

Question 4: MIPS data path

Below we show a simplified version of the 5-stage pipelined MIPS data path (excluding control, forwarding, etc.).



This MIPS supports the base+offset data memory addressing mode, as used in e.g. `LW $1,-8($2)`. Some simple processors do not support this addressing mode, they support only data memory base addressing, as in e.g. `LW $1,($2);` so no offset can be specified. Let's call this processor MIPS-s.

a) How can MIPS support base addressing, as used in MIPS-s, with a single instruction?
b) How can MIPS-s support base+offset addressing: show the MIPS-s instruction equivalent for the MIPS `LW $1,-8($2)` instruction.
c) Since MIPS-s does not support the more complex base+offset addressing mode its pipelining can be simplified to 4 stages, without stretching the cycle time; how? Describe what has to be changed in above data path.
d) The CPI of MIPS-s will be smaller than the CPI of MIPS (for the same compiled C program). Why?
e) How does the execution time of MIPS-s compare to the CPI of MIPS (for the same compiled C program) ? Why?

Answer

a) using offset 0

b) using 2 instructions:
`ADDI $3, $2, -8`
`LW $1, ($2)`

c) the DATA memory functionality can be put in stage 3, next to the ALU and the branch adder. Forwarding logic can be simplified
d) MIPS-s does not have the load lock problem. So on average load take fewer cycles.
e) Since many loads now take 2 instructions (usually 2 cycles) the MIPS-s execution time is longer. In the MIPS-s some loads effectively take only 1 cycle, in case the delay slot can be effectively filled (which is often the case).

Question 5: Caches

We consider a pipelined **64-bit** MIPS processor with a CPI_base = 1.5 . This base CPI assumes ideal memory; all memory accesses are then single cycle cache hits. This processor uses 42 bits addresses for instructions and data. It is connected to both an 32 kByte level-1 instruction cache and 32 kByte **level-1** data cache. Both caches are 4-way set-associative and support single cycle access.
Hit rates for these instruction and data caches are 95 % and 90 % respectively. Both caches use a block size of 4 words (a **word contains 8 bytes**). To fill a cache block from external memory takes 120 cycles. The instruction stream contains 10% stores and 20% loads.

a) The caches use LRU policies. Explain short what this policy means?
b) What is the disadvantage of an associative cache compared to a direct mapped cache?
c) How big is the cache index and cache tag for these level-1 caches?
d) What is the CPI when taking cache misses into account?

To reduce CPI we add a 2 MByte **level-2** cache (shared between instruction and data). This cache is 8-way set-associative, has a block size of 4 words, and 20 cycles access latency. Its hit rate is 75 %.

e) Why is the hit rate of this large level-2 cache lower than the hit rate of the much smaller level-1 caches?
f) What is the CPI when adding this level-2 cache?

Answer

a) Least Recently Used replacement, i.e. when allocating a block the block in the set with the least recently used data is replaced.
b) Have to access all ways in parallel; this requires more energy / access more complex replacement (e.g. using LRU algorithm) extra mux, to select one of the ways, makes access time longer
c) Each way is 8 kByte, containing 8 kByte/(32 Bytes/Block) = 256 entries => **index** = 8 bits **Tag size** = 42 – index – word offset – byte offset = 42 – 8 – 2 – 3 = 29 bits
d) **CPI** = CPI_base + f_instr*f_miss_instr*miss_cycles + f_data*f_miss_data*miss_cycles = 1.5 + 1*0.05*120 + 0.3*0.10*120 = 1.5 + 6 + 3.6 = 11.1 cycles/instruction
e) Because the easy accesses (with much locality) already hit in level-1 and do not reach level-2.
f) **CPI** = CPI_base + f_instr*f_miss_instr_L1*miss_cycles_L1 + f_data*f_miss_data_L1*miss_cycles_L1, where <**miss_cycles_L1>** = hit_time_L2 + miss_rate_L2*access_cycles_external_memory = 20 + 0.25*120 = 50 cycles CPI = 1.5 + 1*0.05*50 + 0.3*0.10*50 = 1.5 + 2.5 + 1.5 = 5.5 cycles/instruction

Question 6: Virtual memory

Consider a virtual memory system with the following properties: 40-bit virtual byte address, 16 kbyte page size, 36-bit physical byte address.

a) Every instruction and data memory access first requires an extra memory access to the page table. This can be avoided by using a TLB. What is a TLB and what does each entry of a TLB contain?
b) What is the total size (in bits) of the page table for each process on this machine, assuming that each entry contains a valid, protection, dirty and use bit, and that all virtual pages are in use. Tricks to reduce the page table are not used.
c) Which techniques can be used to reduce the size of the page table?

Answer

a) A TLB is a cache for address translations; each entry contains the physical address for the indexed virtual address, and access rights (it's a copy of the corresponding page table entry.
b) Page index = 14 bits => Page Table Size = 2^26 * (22+4) bit = 208 Mbyte (or 1664 Mbit)
c)- Dynamic allocation of page table entries - Hashing: inverted page table; 1 entry per physical available instead of virtual page - Page the page table itself (i.e. part of it can be on disk) - Use larger page size (multiple page sizes)

Question 7: Parallel processing and roofline model

We discussed several parallel architectures, exploiting several types of parallism.

a) What is the main type of parallelism supported by GPUs (Graphic Processing Units)?
b) Why is a GPU more energy efficient than a CPU (for a program which maps well to both processors)?
c) How can processors communicate in a shared memory MIMD system, i.e., which operations are used for communicating data?
d) What is the advantage of a ring network (topology) versus a bus, both connecting 8 processors? What is its disadvantage?
e) Draw a typical roofline diagram for a processor; indicate what which quantities are shown along the horizontal and vertical axis.

Answer

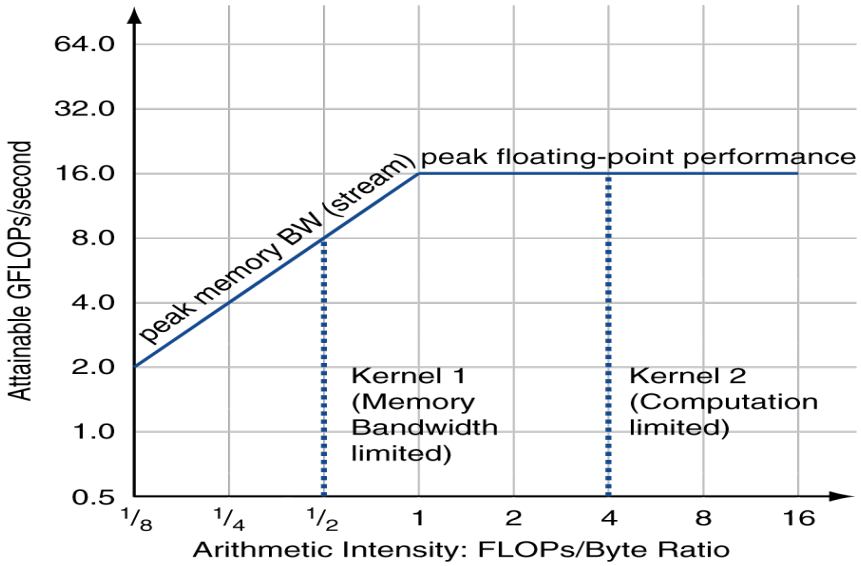
a) SIMD / Vector / Data level parallelism or SIMT

b) SIMD is more efficient since for a whole vector of operations a single instruction has to be fetched and decoded.

c) Just using Load and Store operations (accessing shared memory locations)

d) A ring supports multiple communications at the same time.
However, a ring is more costly. Also a bus makes implementing shared memory, which coherence (e.g. snooping) easier.

e)



Question 8: Operating system
In order to allow correct communication and synchronization between a producer() and a consumer() process, the following code is used. Note that N and the 3 semaphores are shared between the two processes.

```
#define N 10      /* buffer size */
semaphore mutex = 1,
          full = 0,
          empty = N;
```

```
void producer(void)
{
  item i;

  while (TRUE){
    produce_item(&i);
    wait(&mutex);
    wait(&empty);
    add_item(&i);
    signal(&full);
    signal(&mutex);
  }
}
```

```
void consumer(void)
{
  item i;

  while (TRUE){
    wait(&mutex);
    wait(&full);
    remove_item(&item);
    signal(&empty);
    signal(&mutex);
    consume_item(&item);
  }
}
```

Above synchronization is wrong and may cause deadlock.

a) Explain how deadlock can occur.
b) Indicate how the deadlock problem can be solved.

Answer

a) Deadlock may occur. E.g. the producer gets the mutex, enters the critical section, but their are no empty locations (empty=0). Since the consumer can not enter the critical section the buffer remains full forever
b) The mutex semaphore should be checked after the empty / full semaphore.

Question 9: Networks
Suppose that 1000 bytes of user data are to be transmitted over a **4-hop** path in a store-and-forward packet-switched network as a series of packets, each containing **p** data bits and **10** header bits. The bit rate of the lines is 10 Mbps and the propagation delay is negligible.

a) What is the total delay to transmit the user data when p=1000 bits?
b) For what range of values for p is the delay to transmit the user data at most 1 milliseconds.

Answer 9

a) $T_p=(1010)/10^7$=101 microsec for sending 1 packet over 1 hop => T_{tot}= ((8000/1000)packets+3 extra hops)*101=1111 microseconds
b) $T_{tot}=T_p\ (N_p+3)=((p+10)/10^7)*((8000/p)+3)<10^{-3}$sec for 44 <= p <= 613; actually one should only consider the divisors of 8000 in this range.

THE END

Computation II: embedded system design (5EIB0)

Started on	Thursday, 11 April 2019, 4:01 PM
State	Finished
Completed on	Thursday, 11 April 2019, 4:03 PM
Time taken	2 mins 22 secs
Grade	**4.90** out of 5.00 (**98**%)

Question 1
Correct
Mark 1.00 out of 1.00

A Mutex Semaphore is used to satisfy the [Mutual Exclusion ▼] property.

The Wait function must be [Atomic ▼].

The Signal function must be [Atomic ▼].

Your answer is correct.

Question 2
Correct
Mark 1.00 out of 1.00

There are three different types of hazards events:

A [ ] hazard occurs when a required resource is busy.

A [ ] hazard occurs when a an instruction needs to wait for a previous one to finish loading or storing a register.

A [ ] hazard occurs when the next instruction is not directly known.

One method of resolving data hazard is called [ ], in which the missing data element is retrieved from internal buffers.

Structural

Data

Control

Your answer is correct.

Question 3
Correct
Mark 1.00 out of 1.00

Select all the statements that are true about caches. Points are added for correct selections, and deducted for incorrect selections.

Forwarding

Select one or more:

ALU | Instruction | Pipeline | Stall | Control | Structural | Data

- a. A larger cache block reduces the miss penalty.
- b. When a cache uses a write-through scheme, the memory is updated when a cache block is replaced.
- c. The use of a larger cache block reduces the amount of tag storage relative to the amount of data storage in the cache. Correct.
- d. To take advantage of spatial locality, a cache must have a block size smaller than two word.
- e. The use of a larger cache block decreases the miss rate. Correct.

Your answer is correct.

Question 4

Correct

Mark 1.00 out of 1.00

Consider the following code:

```
semaphore mutex = 1;
shared_variable = 0;

/* Thread 1 */
while(TRUE)
{
        wait(&mutex);
        shared_variable = 1;
        signal(&mutex);
}

/* Thread 2 */
while(TRUE)
{
        wait(&mutex);
        shared_variable = 2;
        signal(&mutex);
}
```

Which properties are NOT satisfied by this solution?

Select one or more:

- [ ] a. Mutual exclusion
- [ ] b. Progress
- [ ] c. Deadlock freedom
- [x] d. Starvation freedom

Your answer is correct.

Question 5

Partially correct

Mark 0.90 out of 1.00

Given this instruction sequence,

```
40hex  sub $11, $2, $4
44hex  and $12, $2, $5
48hex  or  $13, $2, $6
4chex  add $1,  $2, $1
50hex  slt $15, $6, $7
54hex  lw  $16, 50($7)
...
```

assume the instructions to be invoked on an exception begin like this:

```
80000180hex  sw   $26, 1000($0)
80000184hex  sw   $27, 1004($0)
...
```

Show what happens in the pipeline during clock cycle 6 and 7, if an overflow exception occurs in the add instruction.

clock 6 IF	lw $16, 50($7) ▼
clock 6 ID	slt $15, $6, $7 ▼
clock 6 EX	add $1, $2, $1 ▼
clock 6 MEM	or $13, $2, $6 ▼
clock 6 WB	and $12, $2, $5 ▼
clock 7 IF	nop ▼
clock 7 ID	nop ▼
clock 7 EX	nop ▼
clock 7 MEM	nop ▼
clock 7 WB	or $13, $2, $6 ▼

Your answer is partially correct.

You have correctly selected 9.

◄ Part 1: 2019-04-02 Practice Exam - Theory

Return to: Practice Assess...

Part 1: 2019-04-02 Practice Exam

https://exam.oncourse.tue.nl/exam/mod/quiz/attempt.php?attempt=4033&cmid=324

Bookmarks | Personal | TU/e | Memes | Project | Steam | Nieuwe ondersteun... | Microsoft support r... | Switch Your Xbox Li... | Motorway route: ca... | Other bookmarks

incourse

Technische Universiteit Eindhoven

oncourse-exam-2018 | My courses | This course | Idse

Dashboard > My > 5EI > Pra > Part 1: 2019-04-02 Practice Exam - Theory

QUIZ NAVIGATION

1 2 3 4 5

Finish attempt ...

Time left **0:27:24**

Computation II: embedded system design (5EIB0)

Question 1

Not yet answered

Marked out of 1.00

Flag question

A Mutex Semaphore is used to satisfy the Choose... property.

The Wait function must be Choose... .

The Signal function must b Choose... .

Choose...
Atomic
Non-Atomic

Next page

Jump to...

Part 2a: 2019-04-02 Mini MIPS Adding Custom Instructions (Binarization Calculation) ▶

Return to: Practice Assess...

ENG 10:02 02/04/2019

QUIZ NAVIGATION

1 2 3 4 5

Finish attempt ...

Time left **0:27:23**

Computation II: embedded system design (5EIB0)

Question 1

Not yet answered

Marked out of 1.00

Flag question

A Mutex Semaphore is used to satisfy the Choose... property.

The Wait function must be Choose... .

The Signal function must be Choose... .

Choose...
Atomic
Non-Atomic

Next page

Jump to...

Part 2a: 2019-04-02 Mini MIPS Adding Custom Instructions (Binarization Calculation) ▶

Return to: Practice Assess...

Part 1: 2019-04-02 Practice Exam
https://exam.oncourse.tue.nl/exam/mod/quiz/attempt.php?attempt=4033&cmid=324&page=1
Bookmarks Personal TU/e Memes Project Steam Nieuwe ondersteun... Microsoft support r... Switch Your Xbox Li... Motorway route: ca... Other bookmarks
incourse
Technische Universiteit Eindhoven
oncourse-exam-2018 My courses This course
Idse

QUIZ NAVIGATION
1 2 3 4 5
Finish attempt ...
Time left **0:27:17**

Computation II: embedded system design (5EIB0)

Question 2
Not yet answered
Marked out of 1.00
Flag question

There are three different types of hazards events:

A [ ] hazard occurs when a required resource is busy.

A [ ] hazard occurs when a an instruction needs to wait for a previous one to finish loading or storing a register.

A [ ] hazard occurs when the next instruction is not directly known.

One method of resolving data hazard is called [ ], in which the missing data element is retrieved from internal buffers.

Data	ALU	Pipeline	Control	Stall	Instruction	Structural	Forwarding

Previous page Next page

Jump to...

Part 2a: 2019-04-02 Mini MIPS Adding Custom Instructions (Binarization Calculation) ▶

Return to: Practice Assess...

ENG 10:02 02/04/2019

QUIZ NAVIGATION

1 2 3 4 5

Finish attempt ...

Time left **0:27:13**

Computation II: embedded system design (5EIB0)

Question 3

Not yet answered

Marked out of 1.00

Flag question

Select all the statements that are true about caches. Points are added for correct selections, and deducted for incorrect selections.

Select one or more:

- a. When a cache uses a write-through scheme, the memory is updated when a cache block is replaced.
- b. A larger cache block reduces the miss penalty.
- c. To take advantage of spatial locality, a cache must have a block size smaller than two word.
- d. The use of a larger cache block reduces the amount of tag storage relative to the amount of data storage in the cache.
- e. The use of a larger cache block decreases the miss rate.

Previous page Next page

Jump to...

Part 2a: 2019-04-02 Mini MIPS Adding Custom Instructions (Binarization Calculation) ▶

Return to: Practice Assess...

QUIZ NAVIGATION
1 2 3 4 5
Finish attempt ...
Time left **0:27:03**

Computation II: embedded system design (5EIB0)

Question 4
Not yet answered
Marked out of 1.00
Flag question

Consider the following code:

```
semaphore mutex = 1;
shared_variable = 0;

/* Thread 1 */
while(TRUE)
{
        wait(&mutex);
        shared_variable = 1;
        signal(&mutex);
}

/* Thread 2 */
while(TRUE)
{
        wait(&mutex);
        shared_variable = 2;
        signal(&mutex);
}
```

Which properties are NOT satisfied by this solution?

Select one or more:

- a. Mutual exclusion
- b. Deadlock freedom
- c. Progress
- d. Starvation freedom

QUIZ NAVIGATION
1 2 3 4 5
Finish attempt ...
Time left **0:26:57**

Computation II: embedded system design (5EIB0)

Question 5
Not yet answered
Marked out of 1.00
Flag question

Given this instruction sequence,

```
40hex  sub $11, $2, $4
44hex  and $12, $2, $5
48hex  or  $13, $2, $6
4chex  add $1,  $2, $1
50hex  slt $15, $6, $7
54hex  lw  $16, 50($7)
...
```

assume the instructions to be invoked on an exception begin like this:

```
80000180hex  sw   $26, 1000($0)
80000184hex  sw   $27, 1004($0)
...
```

Show what happens in the pipeline during clock cycle 6 and 7, if an overflow exception occurs in the add instruction.

clock 6 IF Choose...
clock 6 ID Choose...
clock 6 EX Choose...
clock 6 MEM Choose...
clock 6 WB Choose...
clock 7 IF Choose...

assume the instructions to be invoked on an exception begin like this:

```
80000180hex  sw   $26, 1000($0)
80000184hex  sw   $27, 1004($0)
...
```

Show what happens in the pipeline during clock cycle 6 and 7, if an overflow exception occurs in the add instruction.

clock 6 IF: Choose...
clock 6 ID: Choose...
clock 6 EX: Choose...
clock 6 MEM: Choose...
clock 6 WB: Choose...
clock 7 IF: Choose...
clock 7 ID: Choose...
clock 7 EX: Choose...
clock 7 MEM: Choose...
clock 7 WB: Choose...

Choose...
nop
and $12, $2, $5
lw $16, 50($7)
add $1, $2, $1
or $13, $2, $6
sw $26, 1000($0)
slt $15, $6, $7

Previous page | Finish attempt ...

QUIZ NAVIGATION

1 2 3 4 5

Finish attempt ...

Time left **0:27:26**

Computation II: embedded system design (5EIB0)

Question 1

Not yet answered

Marked out of 1.00

Flag question

A Mutex Semaphore is used to satisfy the Choose... property.

Choose...
Deadlock Freedom
Mutual Exclusion
Starvation Freedom
Progress

The Wait function must be Choose...

The Signal function must be Choose... .

Next page

Jump to...

Part 2a: 2019-04-02 Mini MIPS Adding Custom Instructions (Binarization Calculation) ▶

Return to: Practice Assess...

Computation II: embedded system design (5EIB0)

Started on	Thursday, 16 April 2020, 1:32 PM
State	Finished
Completed on	Thursday, 16 April 2020, 3:17 PM
Time taken	1 hour 44 mins
Overdue	4 mins 26 secs
Grade	**3.25** out of 13.00 (**25**%)

Information

Please note that the programming assignment closing times are set to the closing time of the time extension group. If you do not have a time extension, the question will still close at the correct time, i.e. 15-minutes earlier than indicated. Be aware that this also applies to any countdown timers associated with the assignment.

The password for the programming assignment (VPL) components is **computation**

vpl sessions:

vpl 2a

vpl 2b

vpl 2c

resources:

- Verilog syntax v1.1File
- Verilog Quick ReferenceFile
- Patterson and Hennessy - MIPS Reference Green CardFile
- mMIPS SchematicFile

Question 1

Incorrect

Mark 0.00 out of 1.00

Select all statements that are true. Points are added for correct selections, and deducted for incorrect selections.

Select one or more:

- [ ] a. The **set up time** is the maximum time that the input must be stable before the clock edge
- [ ] b. The **set up time** is the maximum time that the input must be stable after the clock edge
- [x] c. The **set up time** is the minimum time that the input must be stable before the clock edge Correct.
- [ ] d. The **set up time** is the minumum time that the input must be stable after the clock edge
- [x] e. The **hold time** is the maximum time that the input must be stable after the clock edge False, there is no limitation on how long the input can be stable after the clock edge.
- [ ] f. The **hold time** is the maximum time that the input must be stable before the clock edge
- [ ] g. The **hold time** is the minimum time that the input must be stable before the clock edge
- [ ] h. The **hold time** is the minimum time that the input must be stable after the clock edge

Your answer is incorrect.

The correct answers are: The **set up time** is the minimum time that the input must be stable before the clock edge, The **hold time** is the minimum time that the input must be stable after the clock edge

Question 2

Incorrect

Mark 0.00 out of 1.00

Select all statements that are true. Points are added for correct selections and deducted for incorrect selections.

Select one or more:

- [x] a. Allowing ALU instructions to write back their result in the 4th stage rather than the 5th stage improves the performance of a MIPS 5-stage pipeline False, it will cause a structural hazard. Pipeline performance depends on the rate of instruction completion, not on the latency of individual instructions
- [x] b. Pipelining does not improve the latency of individual instructions Correct
- [x] c. Name dependencies such as Write-After-Read and Write-After-Write cause hazard in the 5-stage pipeline and require special handling False, because the register write-back is done in the last stage of the pipeline and is done in program order.
- [x] d. Control hazards are caused by jump and branch instructions that are delayed in a pipelined datapath Correct.
- [ ] e. A pipelined datapath must have separate instruction and data memories as the format of instructions is different from the format of data
- [ ] f. In the MIPS 5-stage pipeline, all Read and Write data hazards can be eliminated by forwarding

Your answer is incorrect.

The correct answers are: Control hazards are caused by jump and branch instructions that are delayed in a pipelined datapath, Pipelining does not improve the latency of individual instructions

Question 3

Correct

Mark 1.00 out of 1.00

In the MIPS pipeline, what are the measures that can be taken to reduce the impact of **data** hazards?

Select one or more:

- [ ] a. Splitting the memory into separate instruction and data memories.
- [ ] b. Replicate the register bank
- [x] c. Implement data forwarding in the datapath.
  Correct.
- [x] d. Allow split register write and read during the two halves of the same clock cycle.
  Correct.

Your answer is correct.

The correct answers are: Implement data forwarding in the datapath., Allow split register write and read during the two halves of the same clock cycle.

Question 4

Correct

Mark 1.00 out of 1.00

Consider an instruction pipeline for the MIPS processor where data references constitute 46% of the instructions, and the ideal CPI ignoring memory structural hazards is 1.05. How much faster is the ideal machine without the memory structural hazard versus the machine with the hazard? Calculate the speedup factor as $T_{exec\text{-}hazard}$ / $T_{exec\text{-}no\text{-}hazard}$, where T is the execution time.

Use at least two decimal figures for your answer.

Answer:

1.438095

Required answer = (Ideal CPI + Data Ref) / Ideal CPI

The correct answer is: 1.4380952380952 $^{+}_{-}$ 0.014380952380952

Question 5

Incorrect

Mark 0.00 out of 1.00

Select the instruction that corresponds to the following instruction code:

000000 01001 01010 11000 00000 100101

(Do not forget that you can use the green card)

Select one:

- a. add s0, s1, t1
- b. sub t8, t9, t0
- c. and t0, t3, t5
- d. j 0x10010100
- e. or t8, t1, t2

Your answer is incorrect.

or t8 t1 t2

OPCODE RS RT RD SHAMT FUNCT

Considering the possible answers, finding the FUNCT code 100101 is enough to answer the question.

The correct answer is: or t8, t1, t2

Question 6

Incorrect

Mark 0.00 out of 1.00

A processor runs at 2 GHz and has a CPI=1.7 for a perfect cache (i.e. without including the stall cycles due to cache misses). Assume that load and store instructions are 20% of the instructions. The processor has an I-cache with a 4% miss rate and a D-cache with 7% miss rate. The hit time is 1 clock cycle for both caches. Assume that the time required to transfer a block of data from the RAM to the cache, the miss penalty, is 15 ns.

What is the number of stall cycles per instruction? Use at least two decimal figures for your answer.

Answer:

0.81

Miss penalty in clock cycles = 15* 10^-9 * 2 * 10^9 (frequency)

Number of stall cycles per instruction = (4/100)*(Miss penalty in clock cycles)+(20/100)*7*(Miss penalty in clock cycles)

The correct answer is: 1.62 $^{+}/_{-}$ 0.1

Question 7

Incorrect

Mark 0.00 out of 1.00

What is the overall CPI? Use at least two decimal figures for your answer.

Answer:

2.51

The correct answer is: 3.32 $^{+}/_{-}$ 0.1

Question 8

Incorrect

Mark 0.00 out of 1.00

What is the average memory access time (AMAT) in ns? Use at least two decimal figures for your answer.

Answer:

12.15

AMAT(IC) = Hit time(IC) + Miss rate(IC) × Miss penalty

AMAT(DC) = Hit time(DC) + Miss rate(DC) × Miss penalty

AMAT = 100/(100+PLS) * AMAT(IC) + PLS/(100+PLS) * AMAT(DC) (PLS = percentage load store)

The correct answer is: 68 $^{+}/_{-}$ 0.1

Question 9

Incorrect

Mark 0.00 out of 0.50

Consider a 2-way set-associative write-back cache with 4096 blocks, where each block has a size of 256 bytes. Memory addresses consist of 32 bits.

Compute number of sets.

$\frac{4096}{2} = 2048$

Answer:

16

Number of Sets = num blocks / blocks per set

The correct answer is: 2048 +/- 0

Question 10

Incorrect

Mark 0.00 out of 0.50

Cache data size (do NOT include the valid, modified, and tag bits) expressed in bytes

$4096 \times 256 = 1048576$

Answer:

26

Cache data size = number of blocks × block size

The correct answer is: 1048576 +/- 0

Question 11

Incorrect

Mark 0.00 out of 0.50

Number of index bits

$2(256 \times 4) = 2084 \Rightarrow \log_2(2084) = 11$

Answer:

2

Index = log2 (# locations)

The correct answer is: 11 +/- 0

Question 12

Incorrect

Mark 0.00 out of 0.50

Number of block offset bits

$\log_2(256) = 8$

Answer:

16

Offset = log2 (block size)

The correct answer is: 8 +/- 0

Question 13

Incorrect

Mark 0.00 out of 0.50

Total valid bits

4096 (1) = 4096

Answer:

1

The correct answer is: 4096 +/- 0

Question 14

Incorrect

Mark 0.00 out of 0.50

Total modified bits

4096

Answer:

1

Equal to number of blocks

The correct answer is: 4096 +/- 0

Question 15

Incorrect

Mark 0.00 out of 0.50

Total number of bits required to store the tag bits in the cache

32 - 11 - 8 = 13

and 13 x 4096 = 53248

Answer:

8

Total number of tag bits = Tag bits x number of blocks

The correct answer is: 53248 +/- 0

Question 16

Partially correct

Mark 0.25 out of 0.50

Following two branch predictors available for a baseline processor. Assume normal CPI is 1, but the branch mispredict penalty is 2 extra stall cycles.

A: 10% misprediction rate, will increase the cycle time by 15%

B: 12% misprediction rate ,will increase the cycle time by 20%

Which predictor would you choose :

If branches are 15% of all instructions A ▼

If branches are 25% of all instructions B ▼

Your answer is partially correct.

1 of your answers is correct.

$CT\_A = 1.15$

$CPI\_A = 0.15 * (0.1 * 3 + 0.9) + (1 - 0.15) = 0.15 * 1.2 + 0.85 = 1.03$

$ET\_A = CT\_A * CPI\_A = 1.15 * 1.03 = 1.185$

$CT\_B = 1.2$

$CPI\_B = 0.15 * (0.12 * 3 + 0.88) + (1 - 0.15) = 0.15 * 1.24 + 0.85 = 1.036$

$ET\_B = CT\_B * CPI\_B = 1.2 * 1.036 = 1.243$

A is the better branch predictor both cases. (second case is not shown in the feedback, but it follows the same calculations)

Question 17

Correct

Mark 1.00 out of 1.00

When running the following code, the output on the terminal is:

```
3061
3061
3062
```

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main(){
        pid_t pid_value;
        printf("%d\n", (int)getpid());

        pid_value = fork();
        if (pid_value!=0){
                printf("%d\n", (int)getpid());
        }
        else{
                printf("%d\n", (int)getpid());
        }
}
```

Which is the parent process ID ?

Answer: 3061

The function *getpid()* returns the unique process id for the current process.

The first message that is printed corresponds to the process id of the initial process, the parent.

Moreover, the parent will print his process id again when entering the first part of the *if* condition, since the variable *pid_value* in the parent process contains the process id of the child, which is different from 0.

The correct answer is: 3061

Question 18

Not answered

Not graded

Use this space to explain any distractions that occurred during the exam. This question is not graded.

◄ Announcements

Part 2a: 2020-04-16 Adding Custom Instructions (Horizontal Gradient Calculation - 1pt) ►

Return to: Final Exam 2020...

Computation II: embedded system design (5EIB0)

Started on	Tuesday, 3 April 2018, 8:49 AM
State	Finished
Completed on	Tuesday, 3 April 2018, 10:36 AM
Time taken	1 hour 46 mins
Grade	**4.27** out of 15.00 (**28**%)

Question 1

Not answered

Not graded

Please enter the number on the USB stick that you are using to take this exam. This question is not graded.

Answer:

Question 2

Incorrect

Mark 0.00 out of 1.50

Consider the 5-stage pipelined MIPS processor, as described in the book, running the following code (9 instructions):

```
lw  $t0, 64($zero)
lw  $t1, 0($t0)
lw  $t2, 4($t0)
add $t5, $t5, 8
add $t3, $t1, $t2
sw  $t3, 12($t0)
lw  $t4, 8($t0)
add $t5, $t1, $t4
sw  $t5, 16($t0)
```

How many cycles are required for the execution of this code when **no forwarding** circuitry has been implemented? Assume **3 stall cycles** in case of RaW dependencies. Omit the 4 cycles required to fill the pipeline.

Answer: 16

Question 3

Incorrect

Mark 0.00 out of 1.50

Consider the 5-stage pipelined MIPS processor, as described in the book, running the following code (7 instructions):

```
 lw  $t1, 0($t0)
 lw  $t2, 4($t0)
add  $t3, $t1, $t2
 sw  $t3, 12($t0)
 lw  $t4, 8($t0)
add  $t5, $t1, $t4
 sw  $t5, 16($t0)
```

How many cycles are required for the execution of this code **with all possible forwarding (bypassing) and remaining hazard detection** circuitry implemented? You may omit the 4 cycles required to fill the pipeline. (Hint: Carefully check all the RaW dependencies).

Answer: 10

Question 4

Incorrect

Mark 0.00 out of 1.00

Assume `$t0` holds the value `0x00101000`.

Consider the following piece of assembly code:

```
      slt  $t2, $zero, $t0
      bne  $t2, $zero, ELSE
      j    DONE
ELSE: addi $t2, $t2, 2
DONE:
```

What is the decimal value of $t2 after these instructions?

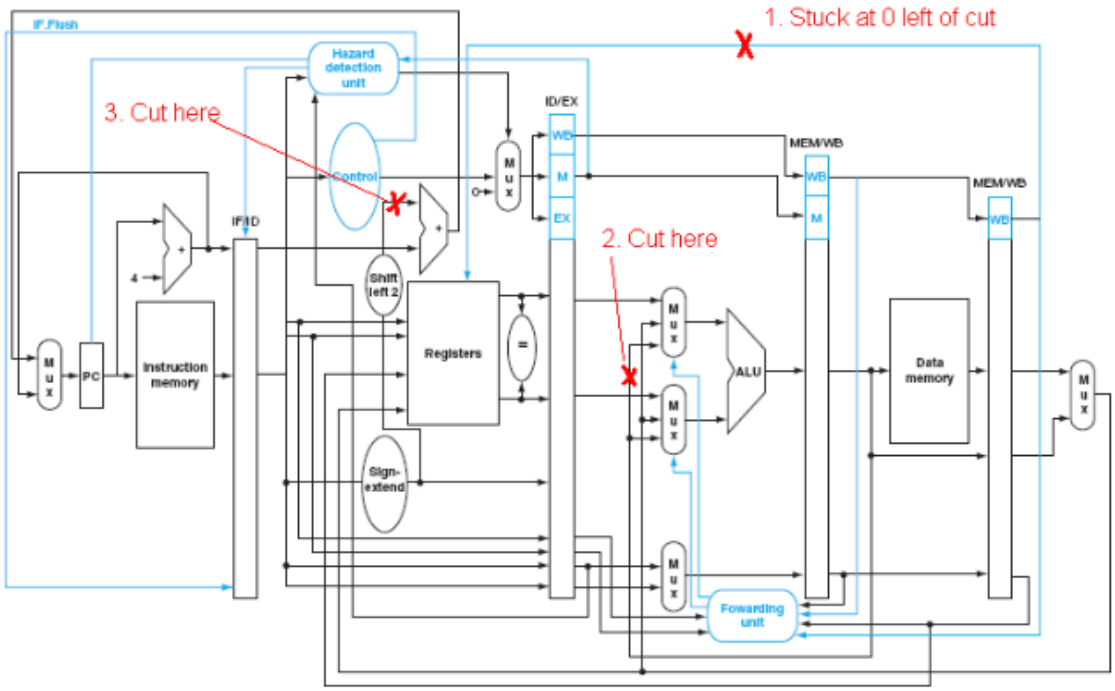
Answer: 40

Question 5

Partially correct

Mark 0.33 out of 1.00

In the data path below we indicate 3 possible errors:



Match the negative consequences to the errors described below **using each answer only once**:

Assuming only error 1: the blue line is cut, and will always be zero left of the cut?

Forwarding of the first operand fails (from MEM to EX)

Assuming only error 2: a cut in a data line?

Cannot write to register file

Assuming only error 3: a cut in a data line?

Jumping to a branch target does not work

Jumping to a branch target does not work

Forwarding of the first operand fails (from MEM to EX)

Cannot write to register file

Question 6

Partially correct

Mark 0.44 out of 1.50

We assume the 5-stage MIPS pipelined processor (from the book, not from the lab). The cycle time of this MIPS depends on its forwarding, as shown below:

Without forwarding	with full forwarding	with ALU->ALU forwarding only
250ps	300ps	290ps

For the following codes, complete the dependencies in front of the codes:

```
Add $1,$2,$3;
Sub $2,$1,$4;   [$1 ▼] [RaW ▼] with [Add ▼]
Add $1,$1,$2;   [$1 ▼] WaW with [Add ▼], [$2 ▼] WaR with
[Sub ▼], [$1 ▼] [WaW ▼] with [Add ▼]
```

Answer the following questions, assuming that the pipeline is already filled.

How many cycles are needed for execution if there is no forwarding? 2

How many cycles are needed for execution if there is full forwarding? 1

How many cycles are needed for execution if there is only ALU → ALU forwarding (no forwarding from the MEM/WB stage to the ALU inputs)? 1

Question 7

Partially correct

Mark 0.50 out of 1.00

The ARM v7 architecture supports an addressing mode not available on MIPS, demonstrated by the next ARM instruction:

```
LDR r1, [r2]; r1=memory[r2]
```

The equivalent MIPS instruction for this ARM instruction would be:

```
LD $1, 0 ($2)
```

Select the possible advantage of this instruction compared to MIPS:

The data memory access could occur in the third pipeline stage ▼

Question 8

Incorrect

Mark 0.00 out of 1.00

Assume a pipelined MIPS architecture with CPI_base = 1.4; this base CPI does not take branch penalties into account. Consider 2 branch predictors: "predict not taken", and a "dynamic" predictor; the dynamic predictor has 80% accuracy. Assume that they both have zero penalty when predicting correctly, and 3 cycles penalty when predicting incorrectly. Branch frequency is 40%.

Calculate the CPI when branches are taken with a 35% frequency, using the "predict not taken" branch predictor and taking branch penalties into account.

Answer:

2.3

Question 9

Incorrect

Mark 0.00 out of 1.50

We consider a pipelined MIPS processor with a CPI_base = 1.4. This base CPI assumes ideal memory; all memory accesses are then single cycle cache hits. This processor uses 40 bits addresses for instructions and data. It is connected to both an 8 kByte level-1 instruction cache and 8 kByte level-1 data cache. Both caches are 4 way set- associative and support single cycle access. Hit rates for these instruction and data caches are 95 % and 90 % respectively. Both caches use a block size of 8 words (a word contains 4 bytes). To fill a cache block from external memory takes 90 cycles. The instruction stream contains 30 % stores and 30 % loads.

The CPI when using this level-1 cache is 11.3 cycles/instruction. To reduce CPI we add a 256 kByte level-2 cache (shared between instruction and data). This cache is also 4-way set associative, has a block size of 8 words, and 20 cycles access latency. Its local hit rate is 60 %.

What is the CPI when using both the level-1 and level-2 caches and taking cache misses into account?

Answer:

15

Question 10

Incorrect

Mark 0.00 out of 1.00

Imagine that you have trained your St. Bernard, Bernie, to carry a box of 9 DVDs instead of a flask of brandy. Each DVD contains 9×10^9 bytes. The dog runs 18 km/hour. What is the maximal distance Bernie can travel to achieve a data rate of at least 4×10^7 bps (bits/sec)? Provide your answer as an integer number in meters.

Answer:

234

Question 11

Correct

Mark 1.00 out of 1.00

The segment of code in which the process may change common variables, update tables, write into files is known as:

Select one:

- a. synchronizing
- b. non – critical section
- c. critical section
- d. program

Question 12

Correct

Mark 1.00 out of 1.00

What are the main differences between threads and processes:

Select one or more:

- [x] a. Threads can communicate via shared variables
- [x] b. Context-switching between processes is more expensive (processing wise) than threads
- [ ] c. None of the above
- [ ] d. Context-switching between threads is more expensive (processing wise) than processes
- [ ] e. Processes can communicate via shared variables
- [ ] f. Both processes and threads can communicate via shared variables
- [ ] g. Threads are protected between each other during context-switching

Question 13

Correct

Mark 1.00 out of 1.00

The time required to create a new thread in an existing process is:

Select one:

- ( ) a. equal to the time required to create a new process
- ( ) b. greater than the time required to create a new process
- (•) c. less than the time required to create a new process
- ( ) d. none of the mentioned

Question 14

Incorrect

Mark 0.00 out of 1.00

Consider a fine-grained multithreaded processor that allows instructions from 2 threads to run concurrently (i.e. there are two function units). Only one instruction per thread can be issued in any cycle.

Assume that we have two threads X and Y that perform the following operations.

Thread X	Thread Y
A1 – takes 3 cycles to execute	B1 – takes 2 cycles to execute
A2 – no dependencies	B2 – conflicts for a functional unit with B1
A3 – conflicts for a functional unit with A1	B3 – depends on the result of B2
A4 - depends on the result of A3	B4 – no dependencies and takes 2 cycles to execute

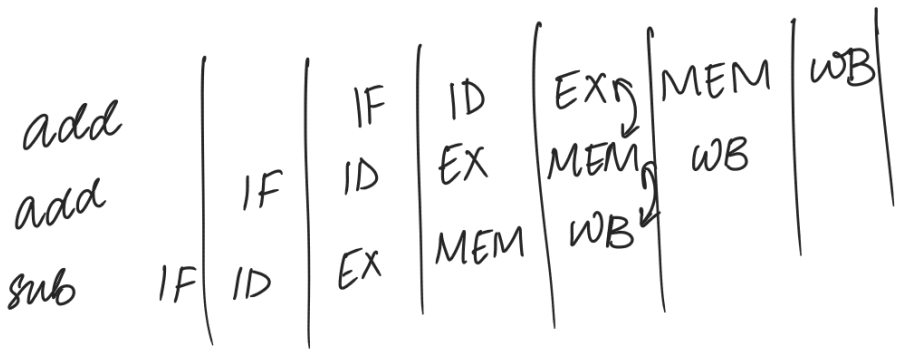
Assuming that you have 1 multithreaded CPU, how many issue slots are wasted, e.g. due to hazards?

Use the following table to help get you started:

FU1	FU2
A1	B1
???	???
???	???

Answer: 4

Return to: Practice Exam 2...


add
add
sub
IF
IF
ID
IF
ID
EX
IF
ID
EX
MEM
IF
ID
EX
MEM
WB
MEM
WB
WB

Computation II: embedded system design (5EIB0)

Started on	Wednesday, 18 April 2018, 1:32 PM
State	Finished
Completed on	Wednesday, 18 April 2018, 4:30 PM
Time taken	2 hours 57 mins
Grade	**10.88** out of 15.00 (**73**%)

Question 1

Correct

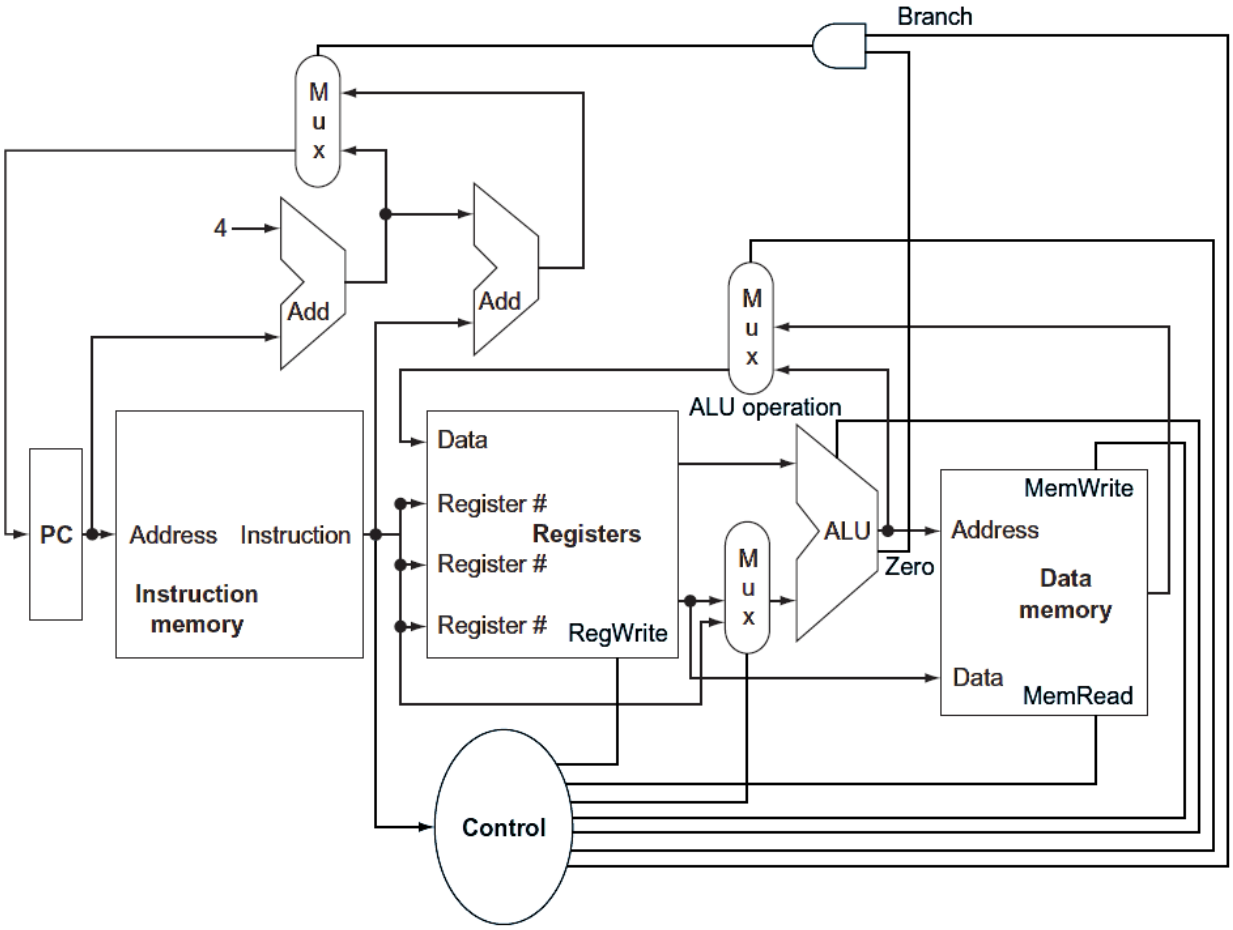
Mark 1.00 out of 1.00

The basic single-cycle MIPS implementation from the textbook shown in the Figure below can only implement some instructions. New instructions can be added to an existing ISA. Consider the new instruction:

```
LWI Rt, Rd(Rs)
```

Interpretation: Reg[Rt] = Mem[Reg[Rd]+Reg[Rs]]

Which existing blocks (if any) can be used for this instruction?



Select one or more:

- [x] a. ALU.
- [x] b. Instruction Memory.
- [ ] c. Branch.
- [x] d. Register Read Ports.
- [ ] e. No existing block can be used for the new instruction.
- [x] f. Register Write Port.
- [x] g. Data Memory.

Question 2

Partially correct

Mark 0.33 out of 0.50

Select the Addressing Modes that are used in the MIPS architectures.

Select one or more:

- [x] a. Register operand (Register addressing)
- [ ] b. Random + Offset (Base or random displacement addressing)
- [ ] c. Register + Offset (Base or displacement addressing)
- [x] d. Immediate operand (Immediate addressing)
- [ ] e. Register + Register (Register displacement addressing)

Question 3
Correct
Mark 0.50 out of 0.50

Consider the R-type instruction format:

op	rs	rt	rd	shamt	funct

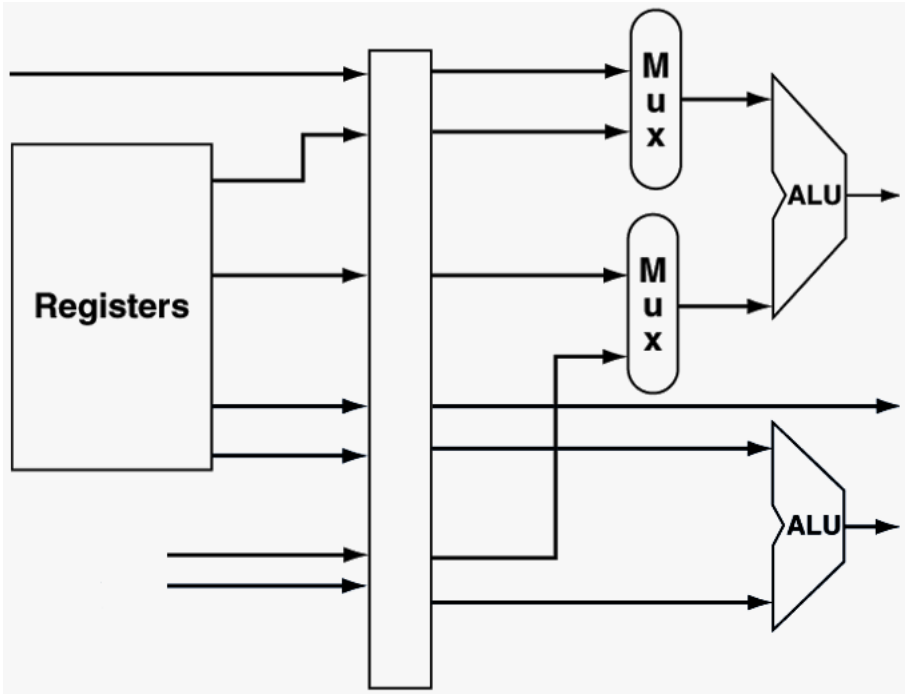
What do *rs, rt* and *rd* mean?

Select one:

- a. Sample, Transfer, and Data register.
- b. Source, Target, and Destination register.
- c. Short, Truncated, and Depth register.
- d. Sink, Temporary, and Data register.

Question 4
Correct
Mark 1.00 out of 1.00

Consider this snippet of a (non-specific) microprocessor architecture:



Assume that all other pipeline stages that are not illustrated can maximally operate on an infinite number of instructions at the same time.

What is the Instruction Per Cycle (IPC) peak in this situation, i.e. what is the maximum number of instructions that can be carried out at the same time? Consider only this stage of the architecture.

Answer: 2

Question 5
Correct
Mark 1.00 out of 1.00

Assume that individual stages of the MIPS data-path have the following latencies:

IF	ID	EX	MEM	WB
200ps	350ps	100ps	600ps	200ps

Furthermore we assume that the instructions executed by the processor have the following distribution:

alu	beq	lw	sw
19%	17%	26%	38%

What is the maximum frequency in a pipelined MIPS processor (in GHz)?

Answer: 1.7

Question 6
Partially correct
Mark 0.25 out of 0.50

There are three different types of hazards events:

A [ ] hazard occurs when a required resource is busy.

A [ ] hazard occurs when a an instruction needs to wait for a previous one to finish loading or storing a register.

A [ ] hazard occurs when the next instruction is not directly known.

One method of resolving data hazard is called [ ], in which the missing data element is retrieved from internal buffers.

Data

Pipeline

Control

Forwarding

Question 7

Incorrect

Mark 0.00 out of 1.00

We assume the 5-stage MIPS pipelined processor (from the book). The cycle time of this MIPS depends on instruction forwarding as follows:

Without forwarding	With full forwarding	With ALU->ALU forwarding only
250ps	300ps	290ps

Consider this sequence of instructions:

```
or r2,r1,r4
or r1,r4,r2
```

Answer the following questions, assuming that the pipeline is empty at the beginning of the instructions. Assume that the necessary number of stall cycle for each hazard is 3.

What is the total execution time (in picoseconds) of this instruction sequence with full forwarding? 600

What is the total execution time (in picoseconds) of this instruction sequence without forwarding? 1250

What is the speedup achieved by adding full forwarding to a pipeline that had no forwarding? 2.08

Question 8

Correct

Mark 0.50 out of 0.50

Consider three branch prediction schemes: predict not taken, predict taken, and dynamic prediction. Assume that they all have zero penalty when they predict correctly and two cycles when they are wrong. Assume that the average predict accuracy of the dynamic predictor is 90%. Which predictor is the best choice for the following branches?

A branch that is taken with 70% frequency: Dynamic prediction

A branch that is taken with 5% frequency: Predict non taken

A branch that is taken with 95% frequency: Predict taken

Question 9

Incorrect

Mark 0.00 out of 1.50

Consider the following sequence of instructions.

```
add r5 , r2 , r1
lw  r3 , 4 (r5)
lw  r2 , 0 (r2)
or  r3 , r5 , r3
sw  r3 , 0 (r5)
```

Assume forwarding and hazard detection is not working properly due to a hasty, end of the term implementation of a student, **how many nops should be inserted** by the programmer to this sequence of instructions to ensure correct execution (the minimum number)? Assume 3 stall cycles.

Answer: 5

Question 10

Partially correct

Mark 0.80 out of 2.00

Given this instruction sequence,

```
40hex  sub $11, $2, $4
44hex  and $12, $2, $5
48hex  or  $13, $2, $6
4chex  add $1,  $2, $1
50hex  slt $15, $6, $7
54hex  lw  $16, 50($7)
...
```

assume the instructions to be invoked on an exception begin like this:

```
80000180hex  sw   $26, 1000($0)
80000184hex  sw   $27, 1004($0)
...
```

Show what happens in the pipeline during clock cycle 6 and 7, if an overflow exception occurs in the add instruction.

clock 6 IF: sw $26, 1000($0) ▼

clock 6 ID: sw $26, 1000($0) ▼

clock 6 EX: add $1, $2, $1 ▼

clock 6 MEM: or $13, $2, $6 ▼

clock 6 WB: and $12, $2, $5 ▼

clock 7 IF: slt $15, $6, $7 ▼

clock 7 ID: sw $26, 1000($0) ▼

clock 7 EX: sw $26, 1000($0) ▼

clock 7 MEM: add $1, $2, $1 ▼

clock 7 WB: or $13, $2, $6 ▼

Question 11

Correct

Mark 0.50 out of 0.50

A Mutex Semaphore is used to satisfy the Mutual Exclusion ▼ property.

The Wait function must be Atomic ▼ .

The Signal function must be Atomic ▼ .

Question 12

Correct

Mark 1.00 out of 1.00

Imagine that you have trained your St. Bernard, Beethoven, to carry a box of 4 Blu-Ray DVDs instead of a flask of brandy. Each Blu-Ray DVD contains 5×10^{10} bytes. The dog runs 36 km/hour. Beethoven is going too fast, and after 10 seconds of running, he needs to stop and rest for 5 seconds (it's speed is 0 km/hour during the break).

What is the average data rate if Beethoven runs for 3800 meters? Provide your answer in bits/second. For simplicity's sake, consider also the last 5 seconds break when the dog reaches its destination.

Answer:

2807017544

Question 13

Correct

Mark 0.50 out of 0.50

Select the correct ordering. The leftmost element is the closest to the processor.

Select one:

- a. L2 Cache - L1 Cache - Hard Drive - DRAM
- b. L1 Cache - L2 Cache - Hard Drive - DRAM
- c. L2 Cache - L1 Cache - DRAM - Hard Drive
- d. L1 Cache - L2 Cache - DRAM - Hard Drive

^

Question 14

Correct

Mark 1.50 out of 1.50

Consider a microprocessor with CPI_{ideal} = 3, an instruction cache Icache and a data cache Dcache.
The Icache miss rate is 2%, while the Dcache miss rate is 7%.
The percentage of load and store instruction is 32%.
The miss penalty is 46 cycles.

Calculate the Slowdown of the processor, in terms of $CPI_{with\ cache\ miss}/CPI_{ideal}$. Use 2 decimal ciphers in your answer (2 becomes 2.00, 3.0151 becomes 3.02).

Answer:

1.65

Question 15

Correct

Mark 2.00 out of 2.00

Consider a microprocessor with CPI_{ideal} = 3, an instruction cache Icache and a data cache Dcache.
The Icache miss rate is 4%, while the Dcache miss rate is 10%.
The percentage of load and store instruction is 36%.
The miss penalty is 48 cycles.

As in the previous question, calculate the Slowdown of the processor, in terms of $CPI_{with\ cache\ miss}/CPI_{ideal}$.

Consider then to double the processor's clock frequency. Calculate the new $CPI_{double\ freq}$ and calculate the Speedup in terms of $T_{exec\text{-}with\ cache\ miss}$ / $T_{exec\text{-}double\ freq}$.
Use 2 decimal ciphers in your answer (2 becomes 2.00, 3.0151 becomes 3.02).

Answer:

1.29

Return to: Final Exam (WIT...

Computation II: embedded system design (5EIB0)

Started on	Wednesday, 3 July 2019, 8:41 PM
State	Finished
Completed on	Wednesday, 3 July 2019, 8:56 PM
Time taken	15 mins 19 secs
Grade	**4.50** out of 12.00 (**38**%)

Question 1

Incorrect

Mark 0.00 out of 1.00

When running the following code, the output on the terminal is:

```
3061
3061
3062
```

original
parent
child

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main(){
        pid_t pid_value;
        printf("%d\n", (int)getpid());

        pid_value = fork();
        if (pid_value!=0){
                printf("%d\n", (int)getpid());
        }
        else{
                printf("%d\n", (int)getpid());
        }
}
```

parent != 0

child = 0

Which is the parent process ID ?

Answer: 3062

3061 ✓

The function *getpid()* returns the unique process id for the current process.

The first message that is printed corresponds to the process id of the initial process, the parent.

Moreover, the parent will print his process id again when entering the first part of the *if* condition, since the variable *pid_value* in the parent process contains the process id of the child, which is different from 0.

The correct answer is: 3061

Question 2

Incorrect

Mark 0.00 out of 1.00

For caches, which of the following are examples of **Temporal Locality**?

Select one or more:

- [ ] a. Instructions in a loop.
- [ ] b. None of the other answers are correct. Choose no other answer.
- [x] c. Array of data.
- [x] d. Single jump (branch) instruction.

Your answer is incorrect.

The correct answer is: Instructions in a loop.

Question 3

Correct

Mark 1.00 out of 1.00

Consider a microprocessor with a CPI_{ideal}, an instruction cache Icache and a data cache Dcache.

The Icache and the Dcache have separate miss rate values. The percentage of load and store instructions is called LD.

What is the correct formula to calculate the CPI considering the cache miss?

Select one:

- a. $CPI_{cache\_miss} = I_{cache\_miss} * miss\_penalty * LD + D_{cache\_miss} * miss\_penalty * LD$
- b. $CPI_{cache\_miss} = I_{cache\_miss} * miss\_penalty + D_{cache\_miss} * miss\_penalty * LD$
- c. $CPI_{cache\_miss} = CPI_{ideal} + I_{cache\_miss} * LD + D_{cache\_miss} * LD$
- d. $CPI_{cache\_miss} = CPI_{ideal} + I_{cache\_miss} * miss\_penalty + D_{cache\_miss} * miss\_penalty * LD$

Your answer is correct.

The correct answer is: $CPI_{cache\_miss} = CPI_{ideal} + I_{cache\_miss} * miss\_penalty + D_{cache\_miss} * miss\_penalty * LD$

Question 4

Correct

Mark 1.00 out of 1.00

Suppose you want to achieve a speed-up of 77 times faster with 113 processors.

What percentage of the original computation must be parallelizable? **Use 4 significant figures in your answer (2 becomes 2.000, 0.015151 becomes 0.01515).**

Answer:

0.9958

From the book at page 505:

Speed-up Challenge **EXAMPLE**

Suppose you want to achieve a speed-up of 90 times faster with 100 processors. What percentage of the original computation can be sequential?

ANSWER

Amdahl's Law (Chapter 1) says

$$\text{Execution time after improvement} = \frac{\text{Execution time affected by improvement}}{\text{Amount of improvement}} + \text{Execution time unaffected}$$

We can reformulate Amdahl's Law in terms of speed-up versus the original execution time:

$$\text{Speed-up} = \frac{\text{Execution time before}}{(\text{Execution time before} - \text{Execution time affected}) + \frac{\text{Execution time affected}}{\text{Amount of improvement}}}$$

This formula is usually rewritten assuming that the execution time before is 1 for some unit of time, and the execution time affected by improvement is considered the fraction of the original execution time:

$$\text{Speed-up} = \frac{1}{(1 - \text{Fraction time affected}) + \frac{\text{Fraction time affected}}{\text{Amount of improvement}}}$$

Substituting 90 for speed-up and 100 for amount of improvement into the formula above:

$$90 = \frac{1}{(1 - \text{Fraction time affected}) + \frac{\text{Fraction time affected}}{100}}$$

The correct answer is: 0.99582560296846 $^{+}_{-}$ 0.0099582560296846

Question 5

Incorrect

Mark 0.00 out of 1.00

Assume that a computer has a network with a **memory bandwidth** of 7 GByte/s.

The **arithmetic intensity** of the kernel is 2 FLOPs/Byte.

The **floating point peak performance** is 16 GFLOPs/s.

What is the **GFLOPs/s** of the computer executing this kernel?

Answer:

30

The computer GFLOPs/s is calculated as the minimum value between the floating point peak performance and the multiplication between the arithmetic intensity and the memory bandwidth (see *Roofline Diagram*)

The correct answer is: 14

Question 6

Correct

Mark 1.00 out of 1.00

Select the assembly instruction that correspond to the following encoding:

```
00100011000010000000000000001111
```

Select one:

- a. add $t1$,t2, $t3
- b. addi $t8$,t0, 15
- c. or $t0$,t9, $t1
- d. sll $t8$,t0, 2

Your answer is correct.

By looking at the Opcode, which are the left-most 6 bits, we understand that the function is an *addi.*

The I-type instruction is divided into:

- 6 bits for the *opcode: 001000 = addi*

- 5 bits for *rs: 11000 = t8*

- 5 bits for *rt: 01000 = t0*

- 16 bits for the *immediate value 0000000000001111 = 15*

The correct answer is: addi $t8$,t0, 15

Question 7

Incorrect

Mark 0.00 out of 1.00

Consider the following instruction:

```
bge $s, $t, address
```

The *Branch if Greater than or Equal* instruction *(bge)* jumps to the instruction at the location *address* if $\$s \geq \$t$.

This is a pseudo code instruction, in practice it will be executed by 2 or more basic instructions.

Which one of the following is a correct implementation of the *bge* instruction?

Select one:

a.

```
slti $at, $s, address
bne $at, $zero, $t
```

b.

```
slt $at, $s, $t
bne $at, $zero, address
```

False, this instruction will jump when s < t.

c.

```
slt $at, $t, $s
bne $at, $zero, address
```

d.

```
slt $at, $s, $t
beq $at, $zero, address
```

Your answer is incorrect.

The correct answer is:

```
slt $at, $s, $t
beq $at, $zero, address
```

Question 8

Not answered

Marked out of 1.00

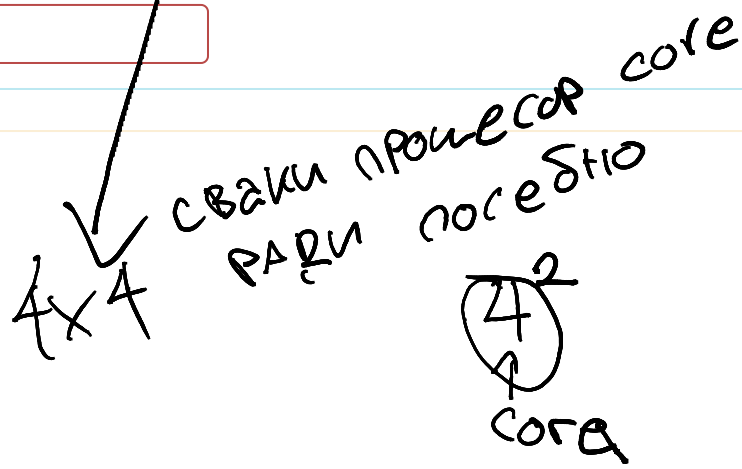
Consider the following portions of two different programs running at the same time on four processors in a symmetric multicore processor (SMP). Assume that before this code is run, both x and y are 0.

Core 1: `x = 1;`
Core 2: `y = 2;`
Core 3: `w = x + y + 1;`
Core 4: `z = x + y;`

If this code was executed a single threaded processor it would result in a single outcome of w = 4 and z = 3. How many unique outcomes can occur for the parallel execution as described above?

Answer:

1. w = 1, z = 0
2. w = 2, z = 0
3. w = 3, z = 0
4. w = 4, z = 0
5. w = 1, z = 1
6. w = 2, z = 1
7. w = 3, z = 1
8. w = 4, z = 1
9. w = 1, z = 2
10. w = 2, z = 2
11. w = 3, z = 2
12. w = 4, z = 2
13. w = 1, z = 3
14. w = 2, z = 3
15. w = 3, z = 3
16. w = 4, z = 3



The correct answer is: 16

Question 9

Incorrect

Mark 0.00 out of 1.00

How are the following parameters affected by changing the frequency of the processor from 50 MHz to 200 MHz?

Write next to each parameter the ratio $Parameter_{200MHz}$ / $Parameter_{50MHz}$. For example, if a value is doubled write 2. If it is halved write 0.5.

Instruction cache miss rate percentage 4

Data cache miss rate percentage 4

Percentage of Load and Store instructions 4

Miss penalty in processor cycles 0

CPI ideal 0

Your answer is incorrect.

The miss penalty corresponds to the time that is necessary to load a new block in the cache from the main memory. The time is not affected by the processor configuration. When increasing the frequency, the same period of time will correspond to a higher number of clock cycles for the processor. In this question, increasing by 4 times the frequency corresponds to a 4x increase in the number of cycles of the miss penalty.

The other parameter are not affected by a change in the frequency.

Question 10

Incorrect

Mark 0.00 out of 1.50

Which of the following are correct ways of implementing a single-cycle No-Operation (NOP or NOOP) instruction?

Select one or more:

- a. *add $t0, $t0, $t0*
  False. You modify the value of the register $t0 and if the register was used in the previous instructions, it may cause a *data hazard*.
- b. *or $t0, $t0, $t0*
  False. If the register $t0 was used in a previous instruction, this may cause a *data hazard.*
- c. sw $zero, 0($t0)
- d. *sll $zero, $zero, 0*
- e. Jump to the next instruction Correct. You add an instruction just to continue normally to the next one.
- f. *lw $t0, 0($t0)*
  False. You load the value MEM[$t0] in $t0, changing its current value.

Your answer is incorrect.

The correct answers are: Jump to the next instruction, *sll $zero, $zero, 0*

Question 11

Correct

Mark 1.50 out of 1.50

Consider a microprocessor with CPI_{ideal} = 4.

Assume that 10% of the instructions are branches.

The percentage of taken branches is 50%.

The penalty when a branch is taken is 2 cycle(s), while when the branch is not taken there is no penalty.

Calculate the Slowdown of the processor, in terms of $CPI_{with\ branches}/CPI_{ideal}$.

Use 2 decimal ciphers in your answer (2 becomes 2.00, 3.0151 becomes 3.02).

Answer:

1.03

The formula for calculating the CPI when considering the branch penalty is (when there is no penalty for non-taken branches):

CPI = CPI_id + % of branch instructions * % of branch taken * branch penalty

The correct answer is: 1.025 +/- 0.01025

$4 + 0,1 \cdot 0,5 \cdot 2 = 4.1$

4.1

4

$\approx 1,025$

Part 2a: 2019-07-03 Adding Custom Instructions (Selective Biasing Calculation - 1pt) ►

Return to: Re-exam 2019-07...

/

Computation II: embedded system design (5EIB0)

Started on	Wednesday, 18 April 2018, 1:33 PM
State	Finished
Completed on	Wednesday, 18 April 2018, 4:36 PM
Time taken	3 hours 2 mins
Grade	**11.03** out of 15.00 (**74**%)

Question 1

Correct

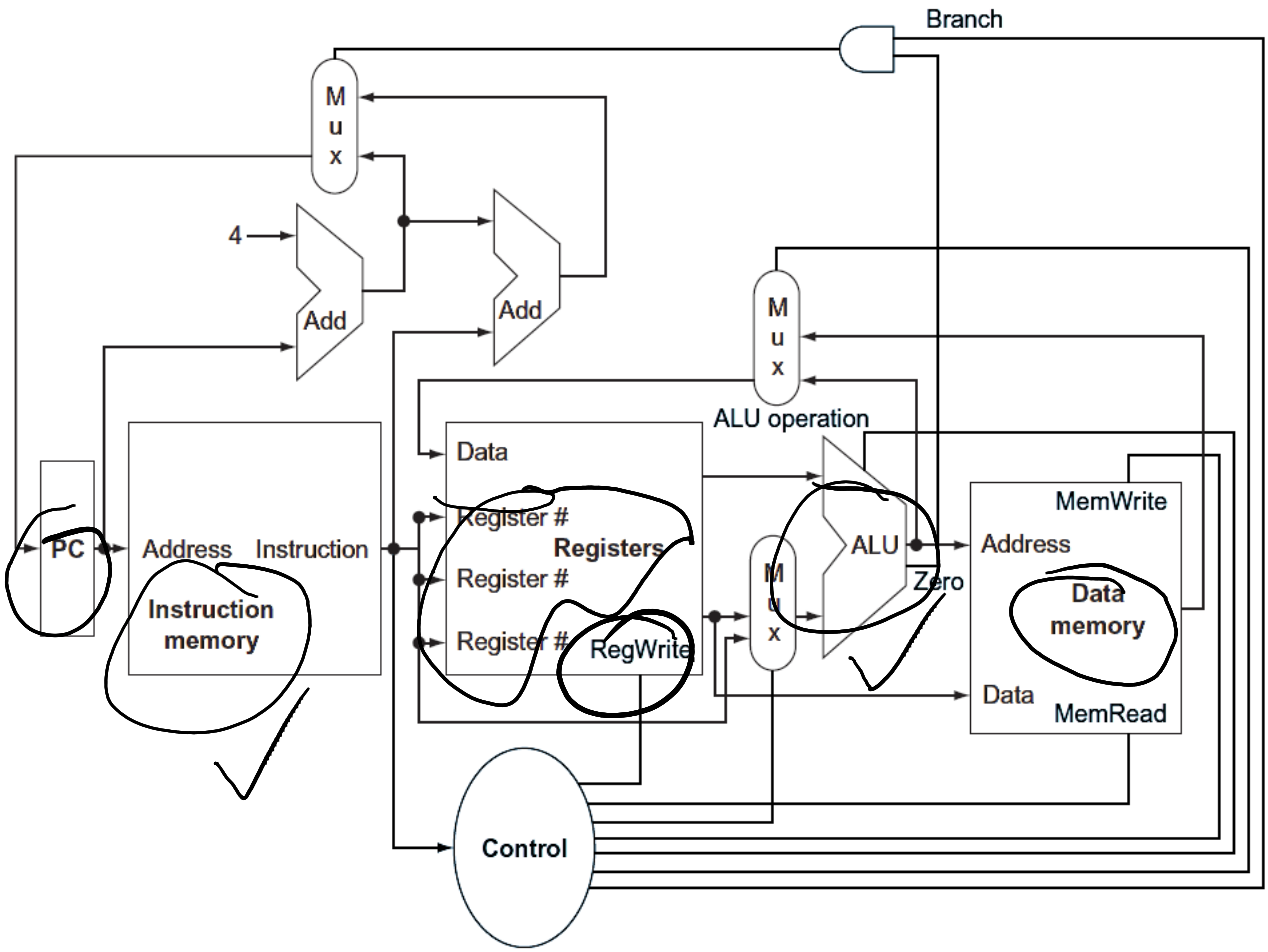
Mark 1.00 out of 1.00

The basic single-cycle MIPS implementation from the textbook shown in the Figure below can only implement some instructions. New instructions can be added to an existing ISA. Consider the new instruction:

```
LWI Rt, Rd(Rs)
```

Interpretation: Reg[Rt] = Mem[Reg[Rd]+Reg[Rs]]

Which existing blocks (if any) can be used for this instruction?



Select one or more:

- [x] a. Instruction Memory.
- [x] b. Data Memory.
- [x] c. ALU.
- [ ] d. No existing block can be used for the new instruction.
- [ ] e. Branch.
- [x] f. Register Write Port.
- [x] g. Register Read Ports.

Question 2

Correct

Mark 0.50 out of 0.50

Select the Addressing Modes that are used in the MIPS architectures.

Select one or more:

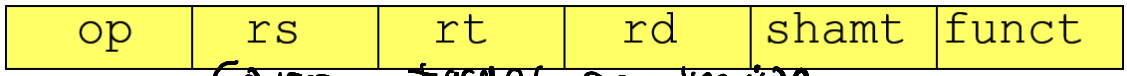
- [x] a. Immediate operand (Immediate addressing)
- [x] b. Register operand (Register addressing)
- [ ] c. Random + Offset (Base or random displacement addressing)
- [x] d. Register + Offset (Base or displacement addressing)
- [ ] e. Register + Register (Register displacement addressing)

Question 3

Correct

Mark 0.50 out of 0.50

Consider the R-type instruction format:



What do *rs*, *rt* and *rd* mean?

Select one:

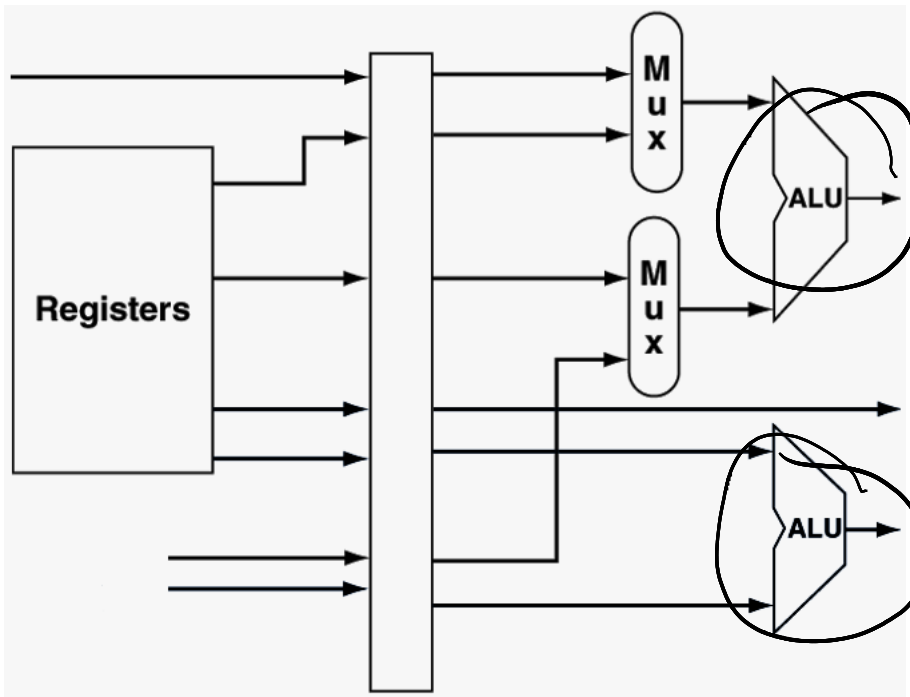
- [ ] a. Sample, Transfer, and Data register.
- [x] b. Source, Target, and Destination register.
- [ ] c. Sink, Temporary, and Data register.
- [ ] d. Short, Truncated, and Depth register.

Question 4

Correct

Mark 1.00 out of 1.00

Consider this snippet of a (non-specific) microprocessor architecture:



Assume that all other pipeline stages that are not illustrated can maximally operate on an infinite number of instructions at the same time.

What is the Instruction Per Cycle (IPC) peak in this situation, i.e. what is the maximum number of instructions that can be carried out at the same time? Consider only this stage of the architecture.

Answer: 2

Question 5

Correct

Mark 1.00 out of 1.00

Assume that individual stages of the MIPS data-path have the following latencies:

IF	ID	EX	MEM	WB
150ps	350ps	200ps	400ps	150ps

Furthermore we assume that the instructions executed by the processor have the following distribution:

alu	beq	lw	sw
19%	18%	21%	42%

What is the maximum frequency in a pipelined MIPS processor (in GHz)?

Answer:

2.5

Question 6

Correct

Mark 0.50 out of 0.50

There are three different types of hazards events:

A Structural hazard occurs when a required resource is busy.

A Data hazard occurs when a an instruction needs to wait for a previous one to finish loading or storing a register.

A Control hazard occurs when the next instruction is not directly known.

One method of resolving data hazard is called Forwarding in which the missing data element is retrieved from internal buffers.

Structural

Data

Control

Forwarding

Question 7

Correct

Mark 1.00 out of 1.00

We assume the 5-stage MIPS pipelined processor (from the book). The cycle time of this MIPS depends on its forwarding, as shown below:

Without forwarding	With full forwarding	With ALU->ALU forwarding only
250ps	300ps	290ps

Consider this sequence of instructions:

```
or r2,r1,r4
or r1,r4,r2
```

Answer the following questions, assuming that the pipeline is empty at the beginning of the instructions. Assume that the necessary number of stall cycle for each hazard is 3.

What is the total execution time (in picoseconds) of this instruction sequence with full forwarding? 1800

What is the total execution time (in picoseconds) of this instruction sequence without forwarding? 2250

What is the speedup achieved by adding full forwarding to a pipeline that had no forwarding? 1.25

Question 8

Correct

Mark 0.50 out of 0.50

Consider three branch prediction schemes: predict not taken, predict taken, and dynamic prediction. Assume that they all have zero penalty when they predict correctly and two cycles when they are wrong. Assume that the average predict accuracy of the dynamic predictor is 90%. Which predictor is the best choice for the following branches?

A branch that is taken with 5% frequency: Predict non taken

A branch that is taken with 95% frequency: Predict taken

A branch that is taken with 70% frequency: Dynamic prediction

Question 9

Correct

Mark 1.50 out of 1.50

Consider the following sequence of instructions.

```
add r1 , r2 , r1
lw  r3 , 4 (r5)
lw  r2 , 0 (r2)
or  r3 , r5 , r3
sw  r3 , 0 (r5)
```

Assume forwarding and hazard detection is not working properly due to a hasty, end of the term implementation of a student, **how many nops should be inserted** by the programmer to this sequence of instructions to ensure correct execution (the minimum number)? Assume 3 stall cycles.

Answer: 8

Question 10

Partially correct

Mark 0.20 out of 2.00

Given this instruction sequence,

```
40hex  sub $11, $2, $4
44hex  and $12, $2, $5
48hex  or  $13, $2, $6
4chex  add $1,  $2, $1
50hex  slt $15, $6, $7
54hex  lw  $16, 50($7)
...
```

assume the instructions to be invoked on an exception begin like this:

```
80000180hex  sw  $26, 1000($0)
80000184hex  sw  $27, 1004($0)
...
```

Show what happens in the pipeline during clock cycle 6 and 7, if an overflow exception occurs in the add instruction.

clock 6 IF	sw $26, 1000($0)
clock 6 ID	sw $26, 1000($0)
clock 6 EX	sw $26, 1000($0)
clock 6 MEM	sw $26, 1000($0)
clock 6 WB	sw $26, 1000($0)
clock 7 IF	sw $26, 1000($0)
clock 7 ID	sw $26, 1000($0)
clock 7 EX	sw $26, 1000($0)
clock 7 MEM	sw $26, 1000($0)
clock 7 WB	sw $26, 1000($0)

Question 11

Partially correct

Mark 0.33 out of 0.50

A Mutex Semaphore is used to satisfy the Mutual Exclusion property.

The Wait function must be Non-Atomic

The Signal function must be Atomic

Question 12

Correct

Mark 1.00 out of 1.00

Imagine that you have trained your St. Bernard, Beethoven, to carry a box of 6 Blu-Ray DVDs instead of a flask of brandy. Each Blu-Ray DVD contains 5×10^{10} bytes. The dog runs 36 km/hour. Beethoven is going too fast, and after 10 seconds of running, he needs to stop and rest for 5 seconds (it's speed is 0 km/hour during the break).

What is the average data rate if Beethoven runs for 3800 meters? Provide your answer in bits/second. For simplicity's sake, consider also the last 5 seconds break when the dog reaches its destination.

Answer:

4210526316

Question 13

Correct

Mark 0.50 out of 0.50

Select the correct ordering. The leftmost element is the closest to the processor.

Select one:

- a. L2 Cache - L1 Cache - Hard Drive - DRAM
- b. L1 Cache - L2 Cache - Hard Drive - DRAM
- c. L2 Cache - L1 Cache - DRAM - Hard Drive
- d. L1 Cache - L2 Cache - DRAM - Hard Drive

Question 14

Correct

Mark 1.50 out of 1.50

Consider a microprocessor with CPI_{ideal} = 4, an instruction cache Icache and a data cache Dcache.
The Icache miss rate is 2%, while the Dcache miss rate is 7%.
The percentage of load and store instruction is 32%.
The miss penalty is 41 cycles.

CompI

Calculate the Slowdown of the processor, in terms of $CPI_{with\ cache\ miss}/CPI_{ideal}$. Use 2 decimal ciphers in your answer (2 becomes 2.00, 3.0151 becomes 3.02).

Answer:

1.43

Question 15

Incorrect

Mark 0.00 out of 2.00

Consider a microprocessor with CPI_{ideal} = 4, an instruction cache Icache and a data cache Dcache.
The Icache miss rate is 2%, while the Dcache miss rate is 7%.
The percentage of load and store instruction is 34%.
The miss penalty is 35 cycles.

CompI

As in the previous question, calculate the Slowdown of the processor, in terms of $CPI_{with\ cache\ miss}/CPI_{ideal}$.

Consider then to double the processor's clock frequency. Calculate the new $CPI_{double\ freq}$ and calculate the Speedup in terms of $T_{exec\text{-}with\ cache\ miss}$ / $T_{exec\text{-}double\ freq}$.
Use 2 decimal ciphers in your answer (2 becomes 2.00, 3.0151 becomes 3.02).

Answer:

2.77

Return to: Final Exam (WIT...

Computation II: embedded system design (5EIB0)

Started on	Thursday, 4 April 2019, 5:19 PM
State	Finished
Completed on	Thursday, 4 April 2019, 5:23 PM
Time taken	4 mins 28 secs
Grade	**5.00** out of 5.00 (**100**%)

4 April 2020

Question 1

Correct

Mark 1.00 out of 1.00

A Mutex Semaphore is used to satisfy the Mutual Exclusion ▼ property.

The Wait function must be Atomic ▼ .

The Signal function must be Atomic ▼ .

Your answer is correct.

Question 2

Correct

Mark 1.00 out of 1.00

There are three different types of hazards events:

A Structural hazard occurs when a required resource is busy.

A Data hazard occurs when a an instruction needs to wait for a previous one to finish loading or storing a register.

A Control hazard occurs when the next instruction is not directly known.

One method of resolving data hazard is called Forwarding in which the missing data element is retrieved from internal buffers.

Structural

Data

Control

Forwarding

Your answer is correct.

Control | Data | Structural | ALU | Pipeline | Stall | Instruction | Forwarding

Question 3

Correct

Mark 1.00 out of 1.00

Select all the statements that are true about caches. Points are added for correct selections, and deducted for incorrect selections.

Select one or more:

- [x] a. The use of a larger cache block decreases the miss rate. Correct.
- [ ] b. To take advantage of spatial locality, a cache must have a block size smaller than two word.
- [ ] c. When a cache uses a write-through scheme, the memory is updated when a cache block is replaced.
- [ ] d. A larger cache block reduces the miss penalty.
- [x] e. The use of a larger cache block reduces the amount of tag storage relative to the amount of data storage in the cache. Correct.

Your answer is correct.

Question 4

Correct

Mark 1.00 out of 1.00

Consider the following code:

```
semaphore mutex = 1;
shared_variable = 0;

/* Thread 1 */
while(TRUE)
{
        wait(&mutex);
        shared_variable = 1;
        signal(&mutex);
}

/* Thread 2 */
while(TRUE)
{
        wait(&mutex);
        shared_variable = 2;
        signal(&mutex);
}
```

Which properties are NOT satisfied by this solution?

Select one or more:

- [ ] a. Deadlock freedom NMA
- [x] b. Starvation freedom
- [ ] c. Progress NMA
- [ ] d. Mutual exclusion NMA

Your answer is correct.

P_0 and P_1 will wait in their respective while loops forever → deadlock

progress: suppose that P_0 and P_1 want to enter the critical section almost simultaneously, and suppose that P_0 is allowed to do so while P_1 busy waits in its **while loop**

IF it is free of deadlocks, have progress and mutual exclusion

Question 5

Correct

Mark 1.00 out of 1.00

Given this instruction sequence,

```
40hex  sub $11, $2, $4
44hex  and $12, $2, $5
48hex  or  $13, $2, $6
4chex  add $1,  $2, $1
50hex  slt $15, $6, $7
54hex  lw  $16, 50($7)
...
```

assume the instructions to be invoked on an exception begin like this:

```
80000180hex  sw   $26, 1000($0)
80000184hex  sw   $27, 1004($0)
...
```

Show what happens in the pipeline during clock cycle 6 and 7, if an overflow exception occurs in the add instruction.

clock 6 IF	lw $16, 50($7) ▼
clock 6 ID	slt $15, $6, $7 ▼
clock 6 EX	add $1, $2, $1 ▼
clock 6 MEM	or $13, $2, $6 ▼
clock 6 WB	and $12, $2, $5 ▼
clock 7 IF	sw $26, 1000($0) ▼
clock 7 ID	nop ▼
clock 7 EX	nop ▼
clock 7 MEM	nop ▼
clock 7 WB	or $13, $2, $6 ▼

Your answer is correct.

◄ Part 1: 2019-04-02 Practice Exam - Theory

Part 2a: 2019-04-02 Mini MIPS Adding Custom Instructions (Binarization Calculation) ►

Return to: Practice Assess...

Computation II: embedded system design (5EIB0)

Started on	Wednesday, 3 July 2019, 3:40 AM
State	Finished
Completed on	Wednesday, 3 July 2019, 6:17 AM
Time taken	2 hours 36 mins

Question 1

Correct

Marked out of 1.00

What is the result in the terminal when running the following code?

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main(){
    pid_t pid_value;
    printf("PID before fork(): %d\n", (int)getpid());

    pid_value = fork();
    if (pid_value!=0){
        printf("Parent. PID = %d\n", (int)getpid());
    }
    else{
        printf("Child. PID = %d\n", (int)getpid());
    }
}
```

Select one:

a.
```
PID before fork(): 1169
Parent. PID = 1170
Child. PID = 0
```

b.
```
PID before fork(): 342
Parent. PID = 342
Child. PID = 343
```

c.
```
PID before fork(): 1315
Parent. PID = 1316
```

d.
```
PID before fork(): 1434
Child. PID = 0
```

Your answer is correct.

When executing the *fork()* function, the return value (in this code the variable *pid_value*) is 0 for the child and the child's PID for the parent.

In this scenario the number printed is not the *pid_value,* but the real PID obtained with the *getpid()* function. Thus the only possible answer is:

Where the printed parent's ID is the same as before the fork, and the child's is the newly assigned PID.

The correct answer is:
```
PID before fork(): 342
Parent. PID = 342
Child. PID = 343
```

Question 2

Correct

Marked out of 1.00

An atomic instruction...

Select one:

- a. ... can be interrupted only by hardware interrupts.
- b. ... is executed indivisibly.
- c. ... is executed in exactly 1 clock cycle.
- d. ... is formed by multiple non-atomic instructions.

Your answer is correct.

The correct answer is: ... is executed indivisibly.

Question 3

Partially correct

Marked out of 1.00

Select all the statements that are **true**.

Select one or more:

- a. In a multiprocessor system, the value read in a cache is always the same as the value read in its corresponding memory address.
- b. The following is a necessary, but not sufficient, condition for coherence: *a read by a processor to location X that follows a write by another processor to X returns the written value if the read and write are sufficiently separated in time and no other writes to X occur between the two accesses.* True.
- c. The following is a necessary, but not sufficient, condition for coherence: *writes to the same location are serialized. Two writes to the same location by any two processors are seen in the same order by all processors.* True.
- d. The following is a necessary, but not sufficient, condition for coherence: *a read by a processor P to a location X that follows a write by P to X, with no other writes of X by another processor occurring between the write and the read by P, may return the written value or the values that was previously contained in X.*
- e. A *write invalidate protocol* can be used to maintain coherence between caches and main memory ensuring that no other readable or writable copies of an item exist when the *write occurs.*

Your answer is partially correct.

You have correctly selected 2.

The correct answers are: A *write invalidate protocol* can be used to maintain coherence between caches and main memory ensuring that no other readable or writable copies of an item exist when the *write occurs.*, The following is a necessary, but not sufficient, condition for coherence: *a read by a processor to location X that follows a write by another processor to X returns the written value if the read* and write are *sufficiently separated in time and no other writes to X occur between the two accesses.*, The following is a necessary, but not sufficient, condition for coherence: *writes to the same location are serialized. Two writes to the same location by any two processors are seen in the same order by all processors.*

Question 4

Correct

Marked out of 1.00

What is the result of the following implementation of the *Producer and Consumer* problem? Each function corresponds to a different thread.

```
# define N 10
semaphore mutex = 1,
          full = 0,
          empty = N;

void producer(void){
    item i;
    while(TRUE){
        produce_item(&i); //Creates a new item
        wait(&full);
        wait(&mutex);
        add_item(&i);     //Adds the new item to a shared buffer
        signal(&mutex);
        signal(&full);
    }
}
void consumer(void){
    item i;
    while(TRUE){
        wait(&empty);
        wait(&mutex);
        remove_item(&i);   //Extracts a produced item from the shared buffer
        signal(&mutex);
        signal(&empty);
        consume_item(&i);  //Consumes the extracted item
    }
```

Select one:

- a. The producer creates and stores in the buffer new items until the buffer is full. The consumer waits forever on the *wait(&empty)*.
- b. Both the producer and the consumer are waiting on their first semaphore forever.
- c. The synchronization works properly: the producer creates new items, stores them in the shared buffer, and the consumer extracts and consumes them.
- d. The producer creates and stores in the buffer a single item, and the consumer extracts and consumes it only once.
- e. The producer creates a new item, then waits forever on the *wait(&full)*. The consumer extracts and consumes non-valid data from the buffer continuously.

Your answer is correct.

The producer should wait on the *empty* semaphore to know if there is space in the buffer. Here instead it is waiting on the *full* semaphore, which is initialized to 0 and no other program is changing its value with a *signal* function. Thus the producer will never add the produced item to the buffer.
The consumer should instead wait on the *full* semaphore, it instead waits on the *empty* semaphore, which is initialized to N (10). The consumer will think that some items are available and will remove them from the (non-initialized) buffer. The value of the extracted item will depend on what was previously written in that memory address. Since it will signal the *empty* semaphore again, it will continuously extract random information from the shared buffer and process them.

The correct answer is: The producer creates a new item, then waits forever on the *wait(&full)*. The consumer extracts and consumes non-valid data from the buffer continuously.

Question 5

Correct

Marked out of 1.00

Select all the statements that are true about **threads**.

Select one or more:

- [x] a. A thread includes the stack.
- [ ] b. None of the other answer is true.
- [x] c. Threads (of the same process) share a single address space.
- [x] d. A thread includes the program counter.
- [x] e. A thread includes the register state.

so when one thread writes some info in mem it is immidiately available to others

Your answer is correct.

The correct answers are: A thread includes the program counter., A thread includes the register state., A thread includes the stack., Threads (of the same process) share a single address space.

A thread includes the program counter, the register state, and the stack.

They commonly share a single address space

Question 6

Correct

Marked out of 1.00

Suppose you want to achieve a speed-up of 78 times faster with 103 processors.

What percentage of the original computation can be sequential? **Use 4 significant figures in your answer (2 becomes 2.000, 0.015151 becomes 0.01515).**

Answer:

0.003142283

From the book at page 505:

Speed-up Challenge **EXAMPLE**

Suppose you want to achieve a speed-up of 90 times faster with 100 processors. What percentage of the original computation can be sequential?

ANSWER

Amdahl's Law (Chapter 1) says

$$\text{Execution time after improvement} = \frac{\text{Execution time affected by improvement}}{\text{Amount of improvement}} + \text{Execution time unaffected}$$

We can reformulate Amdahl's Law in terms of speed-up versus the original execution time:

$$\text{Speed-up} = \frac{\text{Execution time before}}{(\text{Execution time before} - \text{Execution time affected}) + \frac{\text{Execution time affected}}{\text{Amount of improvement}}}$$

This formula is usually rewritten assuming that the execution time before is 1 for some unit of time, and the execution time affected by improvement is considered the fraction of the original execution time:

$$\text{Speed-up} = \frac{1}{(1 - \text{Fraction time affected}) + \frac{\text{Fraction time affected}}{\text{Amount of improvement}}}$$

Substituting 90 for speed-up and 100 for amount of improvement into the formula above:

$$90 = \frac{1}{(1 - \text{Fraction time affected}) + \frac{\text{Fraction time affected}}{100}}$$

The correct answer is: 0.00314228255405 +/- 3.14228255405E-5

$$78 = \frac{1}{(1 - Ffa) + \frac{Ffa}{103}} = \frac{1}{1 - \frac{102}{103} Ffa}$$

$$78 - \frac{78 \cdot 102}{103} Ffa = 1$$

$$\frac{78 - 1}{77.2427} = Ffa$$

$$\Rightarrow Ffa = 0.9968577$$

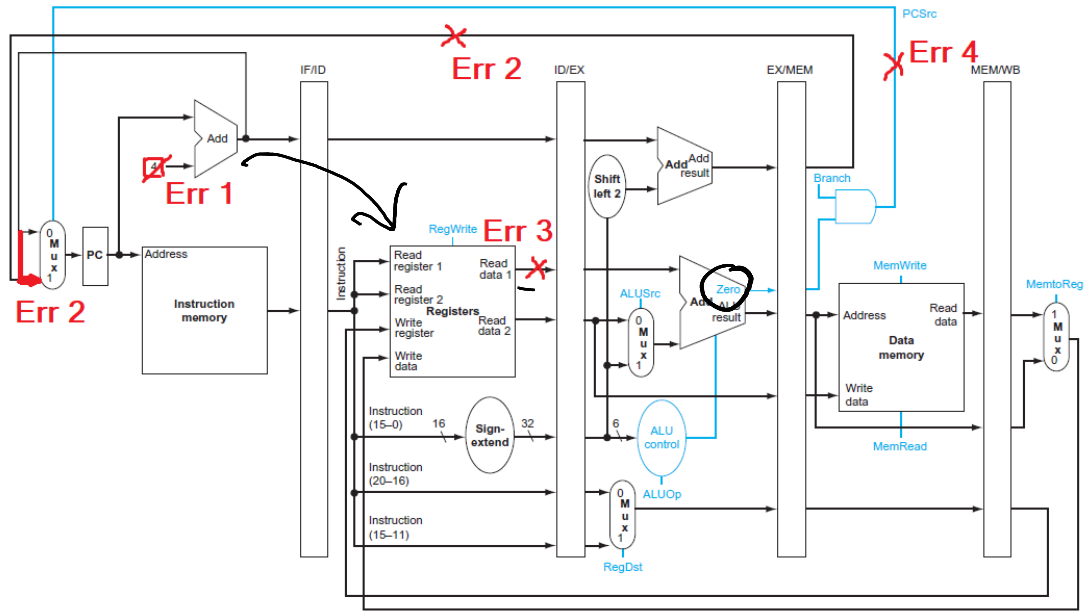
Sequential frac $= 1 - Ffa = 0,00314$

Question 7

Correct

Marked out of 1.00

Match the following faults with their effect on the microcontroller. Use each description only once. Assume that only 1 fault at the time happens.




```
Err 1: The value of the wire is 2 instead of 4.
Err 2: The top wire is disconnected, the left wire reaches both inputs of the
multiplexer.
Err 3: The wire has a stuck at 0 fault.
Err 4: The wire is disconnected, the connection to the control of the multipl
exer is open.
```

Err 1:	Unaligned memory accesses.
Err 2:	The instructions will only be executed sequentially.
Err 3:	One of the ALU operands is always 0.
Err 4:	The next fetched instruction is randomly selected between the following one and the one at the branching address.

- The instructions will only be executed sequentially.
- Unaligned memory accesses.
- One of the ALU operands is always 0.
- The next fetched instruction is randomly selected between the following one and the one at the branching address.

Your answer is correct.

Question 8

Correct

Marked out of 1.00

We assume the 5-stage MIPS pipelined processor (from the book). The cycle time of this MIPS depends on its forwarding, as shown below:

Without forwarding	With full forwarding	With ALU->ALU forwarding only
250ps	300ps	290ps

Consider this sequence of instructions:

```
or r2,r1,r4
or r1,r4,r2
and r5, r1, r2
```

Answer the following questions, assuming that the pipeline is empty at the beginning of the instructions. Assume that the necessary number of stall cycle for each hazard is 3.

What is the total execution time (in picoseconds) of this instruction sequence with full forwarding?

2100

What is the total execution time (in picoseconds) of this instruction sequence without forwarding? 3250

What is the speedup achieved by adding full forwarding to a pipeline that had no forwarding? 1.55

Your answer is correct.

Number of cycles without forwarding = 4 (fill the pipeline) + 3 (instructions) + 6 (stall cycles) = 13

Number of cycles with forwarding = 4 + 3 = 7

Question 9

Correct

Marked out of 1.00

Calculate the Average Memory Access Time (AMAT) for a processor with 3 ns cycle time, a miss penalty of 70 clock cycles, a miss rate of 1 % per instruction, and a cache access time (including hit detection) of 3 clock cycles. Assume that the read and write miss penalties are the same and ignore other write stalls.

Write your result in terms of ns and use 2 decimal ciphers in your answer (2 becomes 2.00, 3.0151 becomes 3.02).

Answer:

11.10

AMATcycles = Time for a hit + Miss rate * Miss penalty (in terms of cycles)

AMATtime = cycles * cycle time

The correct answer is: 11.1 +/- 0.1

$AMAT_{cycles} = 3 + 0,01 \cdot 70 = 3,7$

$AMAT_{time} = 3,7 \cdot 3 = 11,10$

Question 10

Correct

Marked out of 1.00

Consider a microprocessor with CPI_{ideal} = 4, an instruction cache Icache and a data cache Dcache.

The Icache miss rate is 4%, while the Dcache miss rate is 7%.

The percentage of load and store instruction is 35%.

The miss penalty is 41 cycles.

Calculate the Slowdown of the processor, in terms of $CPI_{with\ cache\ miss}/CPI_{ideal}$. Use 2 decimal ciphers in your answer (2 becomes 2.00, 3.0151 becomes 3.02).

Answer:

1.661125

The correct answer is: 1.661125 +/- 0.1

Question 11

Incorrect

Marked out of 1.00

Consider a microprocessor with CPI_{ideal} = 2, an instruction cache Icache and a data cache Dcache.

The Icache miss rate is 4%, while the Dcache miss rate is 10%.

The percentage of load and store instruction is 34%.

The miss penalty is 45 cycles.

As in the previous question, calculate the Slowdown of the processor, in terms of $CPI_{with\ cache\ miss}/CPI_{ideal}$

Consider then to double the processor's clock frequency. Calculate the new $CPI_{double\ freq}$ and calculate the Speedup in terms of $T_{exec\text{-}with\ cache\ miss} / T_{exec\text{-}double\ freq}$. **Submit the Speedup as your answer.**

Use 2 decimal ciphers in your answer (2 becomes 2.00, 3.0151 becomes 3.02).

Answer:

1.804944125

The correct answer is: 1.2309468822171 +/- 0.1

Question 12

Correct

Marked out of 1.00

Consider the following sequence of instructions.

```
add r5 , r2 , r1
lw  r3 , 4 (r5)
lw  r2 , 0 (r2)
or  r3 , r5 , r3
sw  r3 , 0 (r5)
```

Assume forwarding and hazard detection is not working properly. The programmer has to manually insert **nops** to properly stall the processor. **How many nops should be inserted** by the programmer to this sequence of instructions to ensure correct execution (the minimum number)? Assume 3 stall cycles.

Answer: 8

r5 add

or r3 lw

sw r3 or

◄ Part 1: 2019-04-17 Final Exam Theory

Part 2a: 2019-04-17 Adding Custom Instructions (Horizontal Gradient Calculation - 1pt) ►

Return to: Final Exam 2019...

Computation II: embedded system design (5EIB0)

Started on	Wednesday, 3 July 2019, 8:41 PM
State	Finished
Completed on	Wednesday, 3 July 2019, 8:56 PM
Time taken	15 mins 19 secs
Grade	**4.50** out of 12.00 (**38**%)

Question 1

Incorrect

Mark 0.00 out of 1.00

When running the following code, the output on the terminal is:

```
3061
3061
3062
```

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main(){
        pid_t pid_value;
        printf("%d\n", (int)getpid());

        pid_value = fork();
        if (pid_value!=0){
                printf("%d\n", (int)getpid());
        }
        else{
                printf("%d\n", (int)getpid());
        }
}
```

Which is the parent process ID ?

Answer: 3062

The function *getpid()* returns the unique process id for the current process.

The first message that is printed corresponds to the process id of the initial process, the parent.

Moreover, the parent will print his process id again when entering the first part of the *if* condition, since the variable *pid_value* in the parent process contains the process id of the child, which is different from 0.

The correct answer is: 3061

Question 2

Incorrect

Mark 0.00 out of 1.00

For caches, which of the following are examples of **Temporal Locality**?

Select one or more:

- [ ] a. Instructions in a loop.
- [ ] b. None of the other answers are correct. Choose no other answer.
- [x] c. Array of data.
- [x] d. Single jump (branch) instruction.

Your answer is incorrect.

The correct answer is: Instructions in a loop.

Question 3

Correct

Mark 1.00 out of 1.00

Consider a microprocessor with a CPI_{ideal}, an instruction cache Icache and a data cache Dcache.

The Icache and the Dcache have separate miss rate values. The percentage of load and store instructions is called LD.

What is the correct formula to calculate the CPI considering the cache miss?

Select one:

- a. $CPI_{cache\_miss} = I_{cache\_miss} * miss\_penalty * LD + D_{cache\_miss} * miss\_penalty * LD$
- b. $CPI_{cache\_miss} = I_{cache\_miss} * miss\_penalty + D_{cache\_miss} * miss\_penalty * LD$
- c. $CPI_{cache\_miss} = CPI_{ideal} + I_{cache\_miss} * LD + D_{cache\_miss} * LD$
- d. $CPI_{cache\_miss} = CPI_{ideal} + I_{cache\_miss} * miss\_penalty + D_{cache\_miss} * miss\_penalty * LD$

Your answer is correct.

The correct answer is: $CPI_{cache\_miss} = CPI_{ideal} + I_{cache\_miss} * miss\_penalty + D_{cache\_miss} * miss\_penalty * LD$

Question 4

Correct

Mark 1.00 out of 1.00

Suppose you want to achieve a speed-up of 77 times faster with 113 processors.

What percentage of the original computation must be parallelizable? **Use 4 significant figures in your answer (2 becomes 2.000, 0.015151 becomes 0.01515).**

Answer:

0.9958

From the book at page 505:

Speed-up Challenge EXAMPLE

Suppose you want to achieve a speed-up of 90 times faster with 100 processors. What percentage of the original computation can be sequential?

ANSWER

Amdahl's Law (Chapter 1) says

$$\text{Execution time after improvement} = \frac{\text{Execution time affected by improvement}}{\text{Amount of improvement}} + \text{Execution time unaffected}$$

We can reformulate Amdahl's Law in terms of speed-up versus the original execution time:

$$\text{Speed-up} = \frac{\text{Execution time before}}{(\text{Execution time before} - \text{Execution time affected}) + \frac{\text{Execution time affected}}{\text{Amount of improvement}}}$$

This formula is usually rewritten assuming that the execution time before is 1 for some unit of time, and the execution time affected by improvement is considered the fraction of the original execution time:

$$\text{Speed-up} = \frac{1}{(1 - \text{Fraction time affected}) + \frac{\text{Fraction time affected}}{\text{Amount of improvement}}}$$

Substituting 90 for speed-up and 100 for amount of improvement into the formula above:

$$90 = \frac{1}{(1 - \text{Fraction time affected}) + \frac{\text{Fraction time affected}}{100}}$$

The correct answer is: 0.99582560296846 $^{+}_{-}$ 0.0099582560296846

Question 5

Incorrect

Mark 0.00 out of 1.00

Assume that a computer has a network with a **memory bandwidth** of 7 GByte/s.

The **arithmetic intensity** of the kernel is 2 FLOPs/Byte.

The **floating point peak performance** is 16 GFLOPs/s.

What is the **GFLOPs/s** of the computer executing this kernel?

Answer:

30

The computer GFLOPs/s is calculated as the minimum value between the floating point peak performance and the multiplication between the arithmetic intensity and the memory bandwidth (see *Roofline Diagram*)

The correct answer is: $14^{+}/_{-}0$

Question 6

Correct

Mark 1.00 out of 1.00

Select the assembly instruction that correspond to the following encoding:

```
00100011000010000000000000001111
```

Select one:

- a. add $t1$,t2, $t3
- b. addi $t8$,t0, 15
- c. or $t0$,t9, $t1
- d. sll $t8$,t0, 2

Your answer is correct.

By looking at the Opcode, which are the left-most 6 bits, we understand that the function is an *addi.*

The I-type instruction is divided into:

- 6 bits for the *opcode: 001000 = addi*

- 5 bits for *rs: 11000 = t8*

- 5 bits for *rt: 01000 = t0*

- 16 bits for the *immediate value 0000000000001111 = 15*

The correct answer is: addi $t8$,t0, 15

Question 7

Incorrect

Mark 0.00 out of 1.00

Consider the following instruction:

```
bge $s, $t, address
```

The *Branch if Greater than or Equal* instruction *(bge)* jumps to the instruction at the location *address* if $\$s \geq \$t$.

This is a pseudo code instruction, in practice it will be executed by 2 or more basic instructions.

Which one of the following is a correct implementation of the *bge* instruction?

Select one:

a.

```
slti $at, $s, address
bne $at, $zero, $t
```

b.

```
slt $at, $s, $t
bne $at, $zero, address
```

False, this instruction will jump when $s <$t.

c.

```
slt $at, $t, $s
bne $at, $zero, address
```

d.

```
slt $at, $s, $t
beq $at, $zero, address
```

Your answer is incorrect.

The correct answer is:

```
slt $at, $s, $t
beq $at, $zero, address
```

Question 8

Not answered

Marked out of 1.00

Consider the following portions of two different programs running at the same time on four processors in a symmetric multicore processor (SMP). Assume that before this code is run, both x and y are 0.

Core 1: `x = 1;`
Core 2: `y = 2;`
Core 3: `w = x + y + 1;`
Core 4: `z = x + y;`

If this code was executed a single threaded processor it would result in a single outcome of w = 4 and z = 3. How many unique outcomes can occur for the parallel execution as described above?

Answer:

1. w = 1, z = 0
2. w = 2, z = 0
3. w = 3, z = 0
4. w = 4, z = 0
5. w = 1, z = 1
6. w = 2, z = 1
7. w = 3, z = 1
8. w = 4, z = 1
9. w = 1, z = 2
10. w = 2, z = 2
11. w = 3, z = 2
12. w = 4, z = 2
13. w = 1, z = 3
14. w = 2, z = 3
15. w = 3, z = 3
16. w = 4, z = 3

The correct answer is: 16

Question 9

Incorrect

Mark 0.00 out of 1.00

How are the following parameters affected by changing the frequency of the processor from 50 MHz to 200 MHz?

Write next to each parameter the ratio $\text{Parameter}_{200MHz}$ / Parameter_{50MHz} . For example, if a value is doubled write 2. If it is halved write 0.5.

Instruction cache miss rate percentage 4

Data cache miss rate percentage 4

Percentage of Load and Store instructions 4

Miss penalty in processor cycles 0

CPI ideal 0

Your answer is incorrect.

The miss penalty corresponds to the time that is necessary to load a new block in the cache from the main memory. The time is not affected by the processor configuration. When increasing the frequency, the same period of time will correspond to a higher number of clock cycles for the processor. In this question, increasing by 4 times the frequency corresponds to a 4x increase in the number of cycles of the miss penalty.

The other parameter are not affected by a change in the frequency.

Question 10

Incorrect

Mark 0.00 out of 1.50

Which of the following are correct ways of implementing a single-cycle No-Operation (NOP or NOOP) instruction?

Select one or more:

- a. *add $t0, $t0, $t0*
  False. You modify the value of the register $t0 and if the register was used in the previous instructions, it may cause a *data hazard*.
- b. *or $t0, $t0, $t0*
  False. If the register $t0 was used in a previous instruction, this may cause a *data hazard.*
- c. sw $zero, 0($t0)
- d. *sll $zero, $zero, 0*
- e. Jump to the next instruction Correct. You add an instruction just to continue normally to the next one.
- f. *lw $t0, 0($t0)*
  False. You load the value MEM[$t0] in $t0, changing its current value.

Your answer is incorrect.

The correct answers are: Jump to the next instruction, *sll $zero, $zero, 0*

Question 11

Correct

Mark 1.50 out of 1.50

Consider a microprocessor with CPI_{ideal} = 4.

Assume that 10% of the instructions are branches.

The percentage of taken branches is 50%.

The penalty when a branch is taken is 2 cycle(s), while when the branch is not taken there is no penalty.

Calculate the Slowdown of the processor, in terms of $CPI_{with\ branches}/CPI_{ideal}$.

Use 2 decimal ciphers in your answer (2 becomes 2.00, 3.0151 becomes 3.02).

Answer:

1.03

The formula for calculating the CPI when considering the branch penalty is (when there is no penalty for non-taken branches):

CPI = CPI_id + % of branch instructions * % of branch taken * branch penalty

The correct answer is: 1.025 ± 0.01025

Part 2a: 2019-07-03 Adding Custom Instructions (Selective Biasing Calculation - 1pt) ►

Return to: Re-exam 2019-07...

/

Computation II: embedded system design (5EIB0)

Started on	Wednesday, 3 July 2019, 3:40 AM
State	Finished
Completed on	Wednesday, 3 July 2019, 6:17 AM
Time taken	2 hours 36 mins

Question 1

Correct

Marked out of 1.00

What is the result in the terminal when running the following code?

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main(){
        pid_t pid_value;
        printf("PID before fork(): %d\n", (int)getpid());

        pid_value = fork();
        if (pid_value!=0){
                printf("Parent. PID = %d\n", (int)getpid());
        }
        else{
                printf("Child. PID = %d\n", (int)getpid());
        }
}
```

Select one:

a.
```
PID before fork(): 1169
Parent. PID = 1170
Child. PID = 0
```

b.
```
PID before fork(): 342
Parent. PID = 342
Child. PID = 343
```

c.
```
PID before fork(): 1315
Parent. PID = 1316
```

d.
```
PID before fork(): 1434
Child. PID = 0
```

Your answer is correct.

When executing the *fork()* function, the return value (in this code the variable *pid_value)* is 0 for the child and the child's PID for the parent.

In this scenario the number printed is not the *pid_value,* but the real PID obtained with the *getpid()* function. Thus the only possible answer is:

Where the printed parent's ID is the same as before the fork, and the child's is the newly assigned PID.

The correct answer is:
```
PID before fork(): 342
Parent. PID = 342
Child. PID = 343
```

Question 2

Correct

Marked out of 1.00

An atomic instruction...

Select one:

- a. ... can be interrupted only by hardware interrupts.
- b. ... is executed indivisibly.
- c. ... is executed in exactly 1 clock cycle.
- d. ... is formed by multiple non-atomic instructions.

Your answer is correct.

The correct answer is: ... is executed indivisibly.

Question 3

Partially correct

Marked out of 1.00

Select all the statements that are **true**.

Select one or more:

- a. In a multiprocessor system, the value read in a cache is always the same as the value read in its corresponding memory address.
- b. The following is a necessary, but not sufficient, condition for coherence: *a read by a processor to location X that follows a write by another processor to X returns the written value if the read and write are sufficiently separated in time and no other writes to X occur between the two accesses.* True.
- c. The following is a necessary, but not sufficient, condition for coherence: *writes to the same location are serialized. Two writes to the same location by any two processors are seen in the same order by all processors.* True.
- d. The following is a necessary, but not sufficient, condition for coherence: *a read by a processor P to a location X that follows a write by P to X, with no other writes of X by another processor occurring between the write and the read by P, may return the written value or the values that was previously contained in X.*
- e. A *write invalidate protocol* can be used to maintain coherence between caches and main memory ensuring that no other readable or writable copies of an item exist when the *write occurs.*

Your answer is partially correct.

You have correctly selected 2.

The correct answers are: A *write invalidate protocol* can be used to maintain coherence between caches and main memory ensuring that no other readable or writable copies of an item exist when the *write occurs.*, The following is a necessary, but not sufficient, condition for coherence: *a read by a processor to location X that follows a write by another processor to X returns the written value if the read and write are sufficiently separated in time and no other writes to X occur between the two accesses.*, The following is a necessary, but not sufficient, condition for coherence: *writes to the same location are serialized. Two writes to the same location by any two processors are seen in the same order by all processors.*

Question 4

Correct

Marked out of 1.00

What is the result of the following implementation of the *Producer and Consumer* problem? Each function corresponds to a different thread.

```
# define N 10
semaphore mutex = 1,
          full = 0,
          empty = N;

void producer(void){
    item i;
    while(TRUE){
        produce_item(&i); //Creates a new item
        wait(&full);
        wait(&mutex);
        add_item(&i);     //Adds the new item to a shared buffer
        signal(&mutex);
        signal(&full);
    }
}
void consumer(void){
    item i;
    while(TRUE){
        wait(&empty);
        wait(&mutex);
        remove_item(&i);   //Extracts a produced item from the shared buffer
        signal(&mutex);
        signal(&empty);
        consume_item(&i);  //Consumes the extracted item
    }
```

Select one:

- a. The producer creates and stores in the buffer new items until the buffer is full. The consumer waits forever on the *wait(&empty).*
- b. Both the producer and the consumer are waiting on their first semaphore forever.
- c. The synchronization works properly: the producer creates new items, stores them in the shared buffer, and the consumer extracts and consumes them.
- d. The producer creates and stores in the buffer a single item, and the consumer extracts and consumes it only once.
- e. The producer creates a new item, then waits forever on the *wait(&full).* The consumer extracts and consumes non-valid data from the buffer continuously.

Your answer is correct.

The producer should wait on the *empty* semaphore to know if there is space in the buffer. Here instead it is waiting on the *full* semaphore, which is initialized to 0 and no other program is changing its value with a *signal* function. Thus the producer will never add the produced item to the buffer.
The consumer should instead wait on the *full* semaphore, it instead waits on the *empty* semaphore, which is initialized to N (10). The consumer will think that some items are available and will remove them from the (non-initialized) buffer. The value of the extracted item will depend on what was previously written in that memory address. Since it will signal the *empty* semaphore again, it will continuously extract random information from the shared buffer and process them.

The correct answer is: The producer creates a new item, then waits forever on the *wait(&full).* The consumer extracts and consumes non-valid data from the buffer continuously.

Question 5

Correct

Marked out of 1.00

Select all the statements that are true about **threads**.

Select one or more:

- [x] a. A thread includes the stack.
- [ ] b. None of the other answer is true.
- [x] c. Threads (of the same process) share a single address space.
- [x] d. A thread includes the program counter.
- [x] e. A thread includes the register state.

Your answer is correct.

The correct answers are: A thread includes the program counter., A thread includes the register state., A thread includes the stack., Threads (of the same process) share a single address space.

Question 6

Correct

Marked out of 1.00

Suppose you want to achieve a speed-up of 78 times faster with 103 processors.

What percentage of the original computation can be sequential? **Use 4 significant figures in your answer (2 becomes 2.000, 0.015151 becomes 0.01515).**

Answer:

0.003142283

From the book at page 505:

Speed-up Challenge EXAMPLE

Suppose you want to achieve a speed-up of 90 times faster with 100 processors. What percentage of the original computation can be sequential?

ANSWER

Amdahl's Law (Chapter 1) says

$$\text{Execution time after improvement} = \frac{\text{Execution time affected by improvement}}{\text{Amount of improvement}} + \text{Execution time unaffected}$$

We can reformulate Amdahl's Law in terms of speed-up versus the original execution time:

$$\text{Speed-up} = \frac{\text{Execution time before}}{(\text{Execution time before} - \text{Execution time affected}) + \frac{\text{Execution time affected}}{\text{Amount of improvement}}}$$

This formula is usually rewritten assuming that the execution time before is 1 for some unit of time, and the execution time affected by improvement is considered the fraction of the original execution time:

$$\text{Speed-up} = \frac{1}{(1 - \text{Fraction time affected}) + \frac{\text{Fraction time affected}}{\text{Amount of improvement}}}$$

Substituting 90 for speed-up and 100 for amount of improvement into the formula above:

$$90 = \frac{1}{(1 - \text{Fraction time affected}) + \frac{\text{Fraction time affected}}{100}}$$

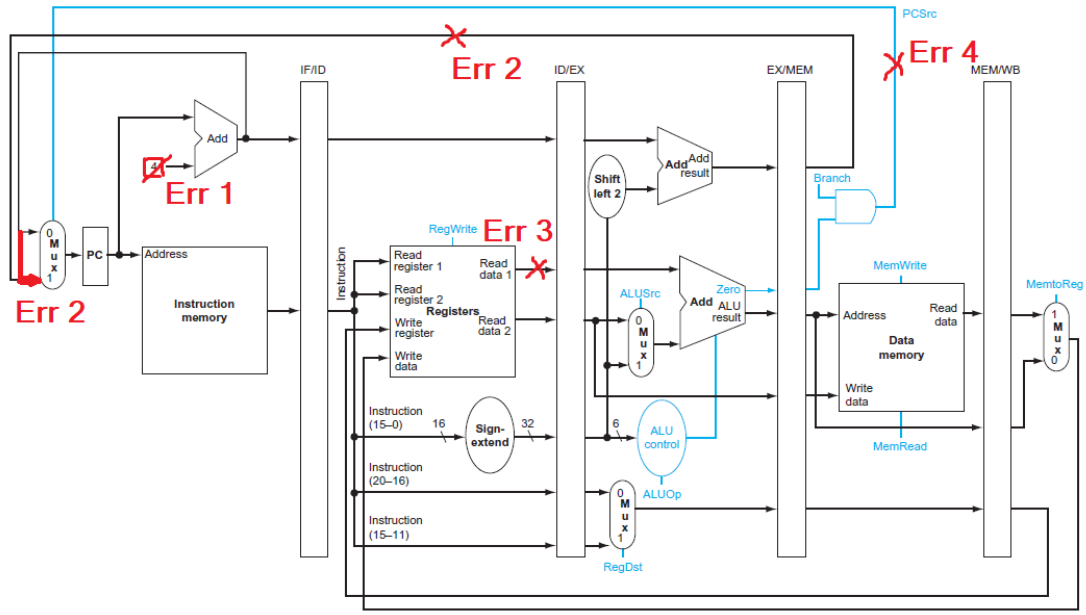
The correct answer is: 0.00314228255405 +/- 3.14228255405E-5

Question 7

Correct

Marked out of 1.00

Match the following faults with their effect on the microcontroller. Use each description only once. Assume that only 1 fault at the time happens.




```
Err 1: The value of the wire is 2 instead of 4.
Err 2: The top wire is disconnected, the left wire reaches both inputs of the multiplexer.
Err 3: The wire has a stuck at 0 fault.
Err 4: The wire is disconnected, the connection to the control of the multiplexer is open.
```

Err 1:	Unaligned memory accesses.
Err 2:	The instructions will only be executed sequentially.
Err 3:	One of the ALU operands is always 0.
Err 4:	The next fetched instruction is randomly selected between the following one and the one at the branching address.

The instructions will only be executed sequentially.

Unaligned memory accesses.

One of the ALU operands is always 0.

The next fetched instruction is randomly selected between the following one and the one at the branching address.

Your answer is correct.

Question 8

Correct

Marked out of 1.00

We assume the 5-stage MIPS pipelined processor (from the book). The cycle time of this MIPS depends on its forwarding, as shown below:

Without forwarding	With full forwarding	With ALU->ALU forwarding only
250ps	300ps	290ps

Consider this sequence of instructions:

```
or r2,r1,r4
or r1,r4,r2
and r5, r1, r2
```

Answer the following questions, assuming that the pipeline is empty at the beginning of the instructions. Assume that the necessary number of stall cycle for each hazard is 3.

What is the total execution time (in picoseconds) of this instruction sequence with full forwarding?

2100

What is the total execution time (in picoseconds) of this instruction sequence without forwarding? 3250

What is the speedup achieved by adding full forwarding to a pipeline that had no forwarding? 1.55

Your answer is correct.

Number of cycles without forwarding = 4 (fill the pipeline) + 3 (instructions) + 6 (stall cycles) = 13

Number of cycles with forwarding = 4 + 3 = 7

Question 9

Correct

Marked out of 1.00

Calculate the Average Memory Access Time (AMAT) for a processor with 3 ns cycle time, a miss penalty of 70 clock cycles, a miss rate of 1 % per instruction, and a cache access time (including hit detection) of 3 clock cycles. Assume that the read and write miss penalties are the same and ignore other write stalls.

Write your result in terms of ns and use 2 decimal ciphers in your answer (2 becomes 2.00, 3.0151 becomes 3.02).

Answer:

11.10

AMATcycles = Time for a hit + Miss rate * Miss penalty (in terms of cycles)
AMATtime = cycles * cycle time

The correct answer is: 11.1 $^{+}/_{-}$ 0.1

Question 10

Correct

Marked out of 1.00

Consider a microprocessor with CPI_{ideal} = 4, an instruction cache Icache and a data cache Dcache.

The Icache miss rate is 4%, while the Dcache miss rate is 7%.

The percentage of load and store instruction is 35%.

The miss penalty is 41 cycles.

Calculate the Slowdown of the processor, in terms of $CPI_{with\ cache\ miss}/CPI_{ideal}$. Use 2 decimal ciphers in your answer (2 becomes 2.00, 3.0151 becomes 3.02).

Answer:

1.661125

The correct answer is: 1.661125 $^{+}/_{-}$ 0.1

Question 11

Incorrect

Marked out of 1.00

Consider a microprocessor with CPI_{ideal} = 2, an instruction cache Icache and a data cache Dcache.

The Icache miss rate is 4%, while the Dcache miss rate is 10%.

The percentage of load and store instruction is 34%.

The miss penalty is 45 cycles.

As in the previous question, calculate the Slowdown of the processor, in terms of $CPI_{with\ cache\ miss}/CPI_{ideal}$.

Consider then to double the processor's clock frequency. Calculate the new $CPI_{double\ freq}$ and calculate the Speedup in terms of $T_{exec\text{-}with\ cache\ miss} / T_{exec\text{-}double\ freq}$. **Submit the Speedup as your answer.**
Use 2 decimal ciphers in your answer (2 becomes 2.00, 3.0151 becomes 3.02).

Answer:

1.804944125

The correct answer is: 1.2309468822171 $^{+}/_{-}$ 0.1

Question 12

Correct

Marked out of 1.00

Consider the following sequence of instructions.

```
add r5 , r2 , r1
lw  r3 , 4 (r5)
lw  r2 , 0 (r2)
or  r3 , r5 , r3
sw  r3 , 0 (r5)
```

Assume forwarding and hazard detection is not working properly. The programmer has to manually insert **nops** to properly stall the processor. **How many nops should be inserted** by the programmer to this sequence of instructions to ensure correct execution (the minimum number)? Assume 3 stall cycles.

Answer: 8

r5 add

or r3 lw

sw r3 or

◄ Part 1: 2019-04-17 Final Exam Theory

Part 2a: 2019-04-17 Adding Custom Instructions (Horizontal Gradient Calculation - 1pt) ►

Return to: Final Exam 2019...

Computation II: embedded system design (5EIB0)

Question 3

Answer saved

Marked out of 1.25

The MIPS processor requires 4-byte word aligned data and instructions. Assume a MIPS processor that jumps to the following address to handle Unaligned Memory exceptions:

`0x80000180 (or 2147484032`$_{\textbf{dec}}$`)`

To what PC address does the following jump instruction jump to?

`0x0800000B`

Provide your answer as a **decimal** number.

(Do not forget that you can use the green card)

Answer:

134217739

Part 2a: 2019-02-22 Finite State Machine (Debugging FSM Divisibility Test) ►

Return to: Practice Interi... ➔

Computation II: embedded system design (5EIB0)

Question 4

Answer saved

Marked out of 1.00

Given the following fragment of MIPS code:

```
sw r16, 12(r6)
lw r16,8(r6)
beq r5,r4,Label # Assume r5!=r4
add r5,r1,r4
slt r5,r15,r4
```

For this question, assume that all the branches are perfectly predicted and that no delay slot is used. If **only one memory is present (for both instructions and data)**, what is the total execution time (in cycles) of this code in the 5-stage pipeline?

Answer: 9

Question 5

Answer saved

Marked out of 0.50

Which mechanism or mechanisms detect the hazards and/or stall the processor in the previous question?

Select all the statements that are true. Points are added for correct selections, and deduced for incorrect selections.

Select one or more:

- [x] a. Branch prediction unit
- [ ] b. Hardware hazard detection unit
- [ ] c. Insert NOPs in the code
- [ ] d. Repeat the fetch stage

Part 2a: 2019-02-22 Finite State Machine (Debugging FSM Divisibility Test) ►

Return to: Practice Interi... ➔

Computation II: embedded system design (5EIB0)

Question 6

Answer saved

Marked out of 1.25

You are tuning the branch prediction method of a 5-stage pipelined MIPS processor to suit a particular application. This application is composed of the following proportion of instructions/types of instruction:

```
R-Type 50%
   BEQ 15%
   JMP  5%
    LW 25%
    SW  5%
```

Having performed some benchmarking of the application for three different types of branch predictor, you find the following branch prediction accuracies:

```
    Always-Taken 30%
Always-Not-Taken 70%
           2-Bit 80%
```

Calculate the exact extra CPI (i.e. do not round the result) due to the mispredicted branches with the Always-Taken predictor. Assume that branch outcomes are determined in the EX stage, that there are no data hazards, and that no delay slots are used.

Answer:

0.0975

Part 2a: 2019-02-22 Finite State Machine (Debugging FSM Divisibility Test) ►

Return to: Practice Interi... 🡒

Computation II: embedded system design (5EIB0)

Question 1

Answer saved

Marked out of 0.50

Select the instruction that corresponds to the following instruction code:

00000 01001 01010 11000 00000 100101

(Do not forget that you can use the green card)

Select one:

- a. add s0, s1, t1
- b. or t1, t2, t8
- c. and t0, t3, t5
- d. j 0x10010100
- e. sub t8, t9, t0

Question 2

Answer saved

Marked out of 0.50

Select the correct combination regarding the Stack and Heap allocation to the memory.

Select one:

- a. The Stack starts after the static memory and grows forwards. The Heap starts after the Stack and grows forwards.
- b. The Stack starts after the Heap and grows forwards. The Heap starts after the static memory and grows forwards.
- c. The Stack starts at the end of the memory and grows backwards. The Heap starts after the Static memory and grows forwards.
- d. The Stack starts after the Static memory and grows forwards. The Heap starts at the end of the memory and grows backwards.

Part 2a: 2019-02-22 Finite State Machine (Debugging FSM Divisibility Test) ►

Return to: Practice Interi... ➔

Computation II: embedded system design (5EIB0)

Started on	Monday, 16 April 2018, 4:40 PM
State	Finished
Completed on	Monday, 16 April 2018, 4:45 PM
Time taken	4 mins 30 secs
Grade	**15.00** out of 15.00 (**100**%)

Question 1

Complete

Not graded

Please enter the number on the USB stick that you are using to take this exam. This question is not graded.

Answer: 123123

Question 2

Correct

Mark 1.50 out of 1.50

Consider the 5-stage pipelined MIPS processor, as described in the book, running the following code (9 instructions):

```
lw  $t0, 64($zero)
lw  $t1, 0($t0)
lw  $t2, 4($t0)
add $t5, $t5, 8
add $t3, $t1, $t2
sw  $t3, 12($t0)
lw  $t4, 8($t0)
add $t5, $t1, $t4
sw  $t5, 16($t0)
```

How many cycles are required for the execution of this code when **no forwarding** circuitry has been implemented? Assume **3 stall cycles** in case of RaW dependencies. Omit the 4 cycles required to fill the pipeline.

Answer: 23

Question 3

Correct

Mark 1.50 out of 1.50

Consider the 5-stage pipelined MIPS processor, as described in the book, running the following code (7 instructions):

```
 lw  $t1, 0($t0)
 lw  $t2, 4($t0)
add  $t3, $t1, $t2
 sw  $t3, 12($t0)
 lw  $t4, 8($t0)
add  $t5, $t1, $t4
 sw  $t5, 16($t0)
```

How many cycles are required for the execution of this code **with all possible forwarding (bypassing) and remaining hazard detection** circuitry implemented? You may omit the 4 cycles required to fill the pipeline. (Hint: Carefully check all the RaW dependencies).

Answer: 9

Question 4

Correct

Mark 1.00 out of 1.00

Assume `$t0` holds the value `0x00101000`.

Consider the following piece of assembly code:

```
      slt  $t2, $zero, $t0
      bne  $t2, $zero, ELSE
      j    DONE
ELSE: addi $t2, $t2, 2
DONE:
```

What is the decimal value of $t2 after these instructions?

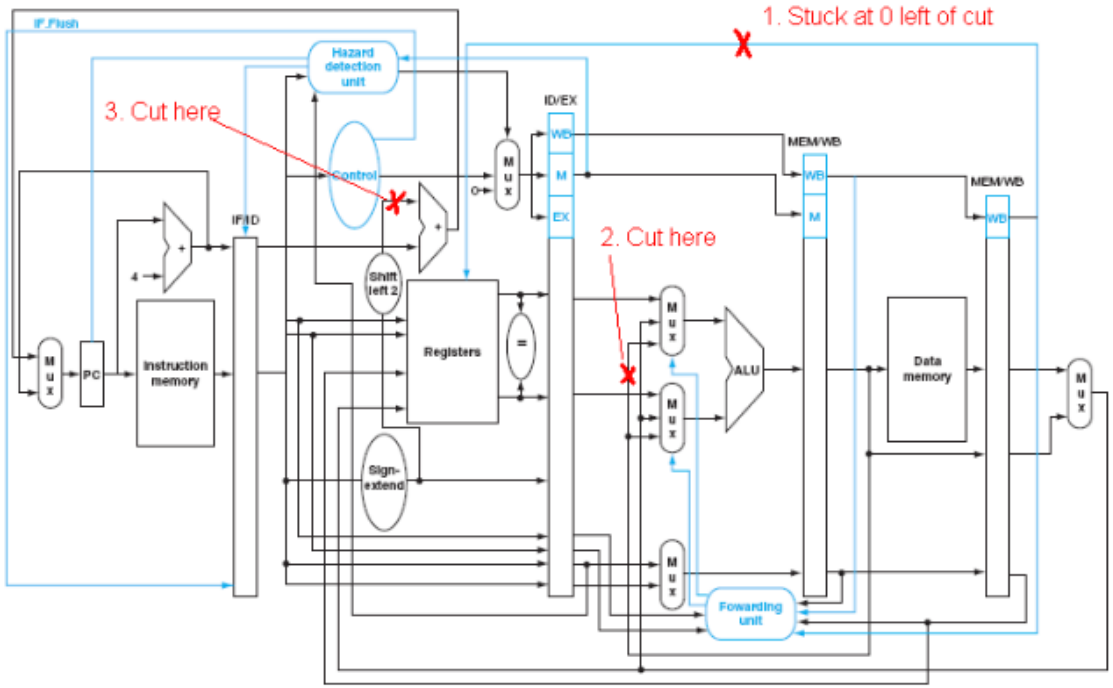
Answer: 3

Question 5

Correct

Mark 1.00 out of 1.00

In the data path below we indicate 3 possible errors:



Match the negative consequences to the errors described below **using each answer only once**:

Assuming only error 1: the blue line is cut, and will always be zero left of the cut?

Assuming only error 2: a cut in a data line?

- Cannot write to register file
- Forwarding of the first operand fails (from MEM to EX)
- Jumping to a branch target does not work
- Cannot write to register file
- Forwarding of the first operand fails (from MEM to EX)

Assuming only error 3: a cut in a data line?

Jumping to a branch target does not work

Question 6

Correct

Mark 1.50 out of 1.50

We assume the 5-stage MIPS pipelined processor (from the book, not from the lab). The cycle time of this MIPS depends on its forwarding, as shown below:

Without forwarding	**with full forwarding**	**with ALU->ALU forwarding only**
250ps	300ps	290ps

For the following codes, complete the dependencies in front of the codes:

```
Add $1,$2,$3;
```

Sub $2,$1,$4; [$1] [RaW] with [Add]

Add $1,$1,$2; [$1] WaW with [Add], [$1] WaR with [Sub], [$2] [RaW] with [Sub]

Answer the following questions, assuming that the pipeline is already filled.

How many cycles are needed for execution if there is no forwarding? 7

How many cycles are needed for execution if there is full forwarding? 3

How many cycles are needed for execution if there is only ALU $\rightarrow$ ALU forwarding (no forwarding from the MEM/WB stage to the ALU inputs)? 4

Question 7

Correct

Mark 1.00 out of 1.00

The ARM v7 architecture supports an addressing mode not available on MIPS, demonstrated by the next ARM instruction:

```
LDR r1, [r2]; r1=memory[r2]
```

The equivalent MIPS instruction for this ARM instruction would be:

```
LD $1, 0 ($2)
```

Select the possible advantage of this instruction compared to MIPS:

All choices are correct ▼

Question 8

Correct

Mark 1.00 out of 1.00

Assume a pipelined MIPS architecture with CPI_base = 1.6; this base CPI does not take branch penalties into account. Consider 2 branch predictors: "predict not taken", and a "dynamic" predictor; the dynamic predictor has 90% accuracy. Assume that they both have zero penalty when predicting correctly, and 3 cycles penalty when predicting incorrectly. Branch frequency is 20%.

Calculate the CPI when branches are taken with a 45% frequency, using the "predict not taken" branch predictor and taking branch penalties into account.

Answer:

1.87

Question 9

Correct

Mark 1.50 out of 1.50

We consider a pipelined MIPS processor with a CPI_base = 1.6. This base CPI assumes ideal memory; all memory accesses are then single cycle cache hits. This processor uses 40 bits addresses for instructions and data. It is connected to both an 8 kByte level-1 instruction cache and 8 kByte level-1 data cache. Both caches are 4 way set- associative and support single cycle access. Hit rates for these instruction and data caches are 95 % and 80 % respectively. Both caches use a block size of 8 words (a word contains 4 bytes). To fill a cache block from external memory takes 50 cycles. The instruction stream contains 30 % stores and 30 % loads.

The CPI when using this level-1 cache is 10.1 cycles/instruction. To reduce CPI we add a 256 kByte level-2 cache (shared between instruction and data). This cache is also 4-way set associative, has a block size of 8 words, and 10 cycles access latency. Its local hit rate is 70 %.

What is the CPI when using both the level-1 and level-2 caches and taking cache misses into account?

Answer:

5.85

Question 10

Correct

Mark 1.00 out of 1.00

Imagine that you have trained your St. Bernard, Bernie, to carry a box of 2 DVDs instead of a flask of brandy. Each DVD contains 9×10^9 bytes. The dog runs 18 km/hour. What is the maximal distance Bernie can travel to achieve a data rate of at least 2×10^7 bps (bits/sec)? Provide your answer as an integer number in meters.

Answer:

36000

Question 11

Correct

Mark 1.00 out of 1.00

The segment of code in which the process may change common variables, update tables, write into files is known as:

Select one:

- a. non – critical section
- b. critical section
- c. program
- d. synchronizing

Question 12

Correct

Mark 1.00 out of 1.00

What are the main differences between threads and processes:

Select one or more:

- [ ] a. None of the above
- [x] b. Context-switching between processes is more expensive (processing wise) than threads
- [ ] c. Both processes and threads can communicate via shared variables
- [ ] d. Processes can communicate via shared variables
- [ ] e. Context-switching between threads is more expensive (processing wise) than processes
- [ ] f. Threads are protected between each other during context-switching
- [x] g. Threads can communicate via shared variables

Question 13

Correct

Mark 1.00 out of 1.00

The time required to create a new thread in an existing process is:

Select one:

- ( ) a. none of the mentioned
- (•) b. less than the time required to create a new process
- ( ) c. greater than the time required to create a new process
- ( ) d. equal to the time required to create a new process

Question 14

Correct

Mark 1.00 out of 1.00

Consider a fine-grained multithreaded processor that allows instructions from 2 threads to run concurrently (i.e. there are two function units). Only one instruction per thread can be issued in any cycle.

Assume that we have two threads X and Y that perform the following operations.

Thread X	Thread Y
A1 – takes 3 cycles to execute	B1 – takes 2 cycles to execute
A2 – no dependencies	B2 – conflicts for a functional unit with B1
A3 – conflicts for a functional unit with A1	B3 – depends on the result of B2
A4 - depends on the result of A3	B4 – no dependencies and takes 2 cycles to execute

Assuming that you have 1 multithreaded CPU, how many issue slots are wasted, e.g. due to hazards?

Use the following table to help get you started:

FU1	FU2
A1	B1
???	???
???	???

Answer: 1

Return to: Practice Exam 2... ➔

Computation II: embedded system design (5EIB0)

Question 4

Answer saved

Marked out of 1.00

Given the following fragment of MIPS code:

```
sw r16, 12(r6)
lw r16,8(r6)
beq r5,r4,Label # Assume r5!=r4
add r5,r1,r4
slt r5,r15,r4
```

For this question, assume that all the branches are perfectly predicted and that no delay slot is used. If **only one memory is present (for both instructions and data)**, what is the total execution time (in cycles) of this code in the 5-stage pipeline?

Answer: 9

Question 5

Answer saved

Marked out of 0.50

Which mechanism or mechanisms detect the hazards and/or stall the processor in the previous question?

Select all the statements that are true. Points are added for correct selections, and deduced for incorrect selections.

Select one or more:

- [x] a. Branch prediction unit
- [ ] b. Hardware hazard detection unit
- [ ] c. Insert NOPs in the code
- [ ] d. Repeat the fetch stage

Part 2a: 2019-02-22 Finite State Machine (Debugging FSM Divisibility Test) ►

Return to: Practice Interi... ➔

Computation II: embedded system design (5EIB0)

Question 6

Answer saved

Marked out of 1.25

You are tuning the branch prediction method of a 5-stage pipelined MIPS processor to suit a particular application. This application is composed of the following proportion of instructions/types of instruction:

```
R-Type 50%
   BEQ 15%
   JMP  5%
    LW 25%
    SW  5%
```

Having performed some benchmarking of the application for three different types of branch predictor, you find the following branch prediction accuracies:

```
    Always-Taken 30%
Always-Not-Taken 70%
           2-Bit 80%
```

Calculate the exact extra CPI (i.e. do not round the result) due to the mispredicted branches with the Always-Taken predictor. Assume that branch outcomes are determined in the EX stage, that there are no data hazards, and that no delay slots are used.

Answer:

0.0975

Part 2a: 2019-02-22 Finite State Machine (Debugging FSM Divisibility Test) ►

Return to: Practice Interi... 🡒

Computation II: embedded system design (5EIB0)

Question 1

Answer saved

Marked out of 0.50

Select the instruction that corresponds to the following instruction code:

00000 01001 01010 11000 00000 100101

(Do not forget that you can use the green card)

Select one:

- a. add s0, s1, t1
- b. or t1, t2, t8
- c. and t0, t3, t5
- d. j 0x10010100
- e. sub t8, t9, t0

Question 2

Answer saved

Marked out of 0.50

Select the correct combination regarding the Stack and Heap allocation to the memory.

Select one:

- a. The Stack starts after the static memory and grows forwards. The Heap starts after the Stack and grows forwards.
- b. The Stack starts after the Heap and grows forwards. The Heap starts after the static memory and grows forwards.
- c. The Stack starts at the end of the memory and grows backwards. The Heap starts after the Static memory and grows forwards.
- d. The Stack starts after the Static memory and grows forwards. The Heap starts at the end of the memory and grows backwards.

Part 2a: 2019-02-22 Finite State Machine (Debugging FSM Divisibility Test) ►

Return to: Practice Interi... ➔

Computation II: embedded system design (5EIB0)

Question 3

Answer saved

Marked out of 1.25

The MIPS processor requires 4-byte word aligned data and instructions. Assume a MIPS processor that jumps to the following address to handle Unaligned Memory exceptions:

`0x80000180 (or 2147484032`$_{\textbf{dec}}$`)`

To what PC address does the following jump instruction jump to?

`0x0800000B`

Provide your answer as a **decimal** number.

(Do not forget that you can use the green card)

Answer:

134217739

Part 2a: 2019-02-22 Finite State Machine (Debugging FSM Divisibility Test) ►

Return to: Practice Interi... ➔