Full Name: _____ NetID: _____

Final Exam 12/07/2016

OAKLAND UNIVERSITY, School of Engineering and Computer Science
CSE 564: Computer Architecture, Fall 2016

Please write and/or mark your answers clearly and neatly; answers that are not readable are not considered. Please answer in appropriate details if needed and you may get partial points for the intermediate steps even if a final answer is not correct.

1. A program has two parallel portions, and one sequential portion. The two parallel portions make 30% and 40% of program respectively. If executing the program on a CPU with 4 cores, the speedups for the two parallel portions are 3 and 4. What is the overall speedup of the program? (4 points)

Speedup = $1/(0.3 + 0.3/3 + 0.4/4) = 2$.

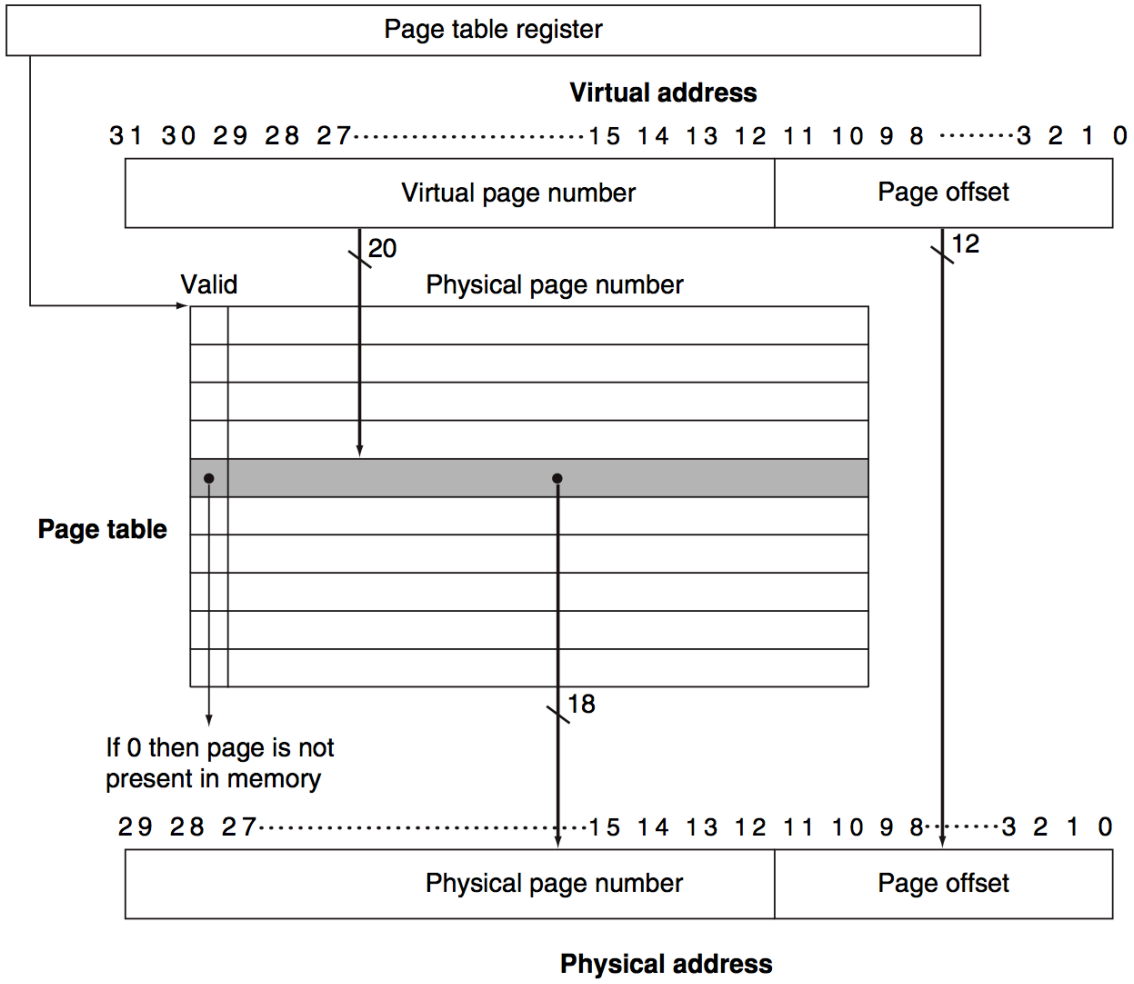
2. Suppose a processor executes at 200MHz clock rate (5 ns per cycle) with ideal CPI = 1. A program contains 40% arith/logic instructions, 50% ld/st, 10% control transfer instructions. If the miss rate for data cache is 0.1 and for instruction cache 0.02, both with 50 cycles miss penalty. What is the average CPI of the program? (6 points)

CPI = ideal CPI + average stalls per instruction = 1(cycles/ins) + [0.50 (DataMops/ins) x 0.10 (miss/DataMop) x 50 (cycle/miss)] + [1 (InstMop/ins) x 0.02 (miss/InstMop) x 50 (cycle/miss)] = (1 + 2.5 + 1) cycle/ins = 4.5

3. 1). Name the four kinds of cache misses and the three goals that most of the cache optimization techniques attempt to improve. 2). What kind of miss prefetching (software or hardware) would reduce the most? 3). In the multiple-level cache design, which goal is the L1-cache designed mainly for? Which goal is the L2 or L3 cache is mainly designed for? (10 points)

1). Misses: Compulsory, conflict, capacity and coherence
Goals: Miss rate, miss penalty and hit time.
2). Compulsory miss
3). L1: reduce hit time; L2/L3: reduce miss rate

4. For following page table configuration for a 32-bit machine that has 4Kb page size and 256Mbyte of physical memory, the page table size (each page table entry is rounded to 2^x bits which is 32 bit) is 4 bytes * $2^{20} = 2^{22}$ bytes = 4Mbytes. What is the page table size if the physical memory is 64Mbytes? (5 points)



64Mbyte/4K = $2^{(26-12)} = 2^{14}$ pages, thus we need 14 bits for physical page number, plus one valid bit, we need 15 bits (rounded up to 16 bits) per page entry. Thus the page table size is

2 bytes * $2^{20} = 2^{21}$ bytes = 2MBytes

5. What is the purpose of TLB? (5 points)

6. Fill in the following table for the three kinds of dependencies (5 points)

1. DIV.D F0, F2, F4
2. ADD.D F6, F0, F8
3. S.D F6, 0(R1)
4. SUB.D F8, F10, F14
5. MUL.D F6, F10, F8

Dependency type	Register involved	1st Instruction	2nd instruction
RAW	F0	1	2
RAW	F6	2	3
RAW	F8	4	5
WAR	F6	3	5
WAW	F6	2	5

The following C code, its corresponding assembly instructions, instruction latency table and the instruction-scheduling table will be used for questions 7 and 8.

```
for (i=999; i>=0; i=i–1)
        x[i] = x[i] + s;
```

```
Loop:   L.D      F0,0(R1)     ;F0=array element
        ADD.D    F4,F0,F2     ;add scalar in F2
        S.D      F4,0(R1)     ;store result
        DADDUI   R1,R1,#-8    ;decrement pointer
                              ;8 bytes (per DW)
        BNE      R1,R2,Loop   ;branch R1!=R2
```

For a machine with instruction latency as follows:

Instruction producing result	Instruction using result	Latency in clock cycles
FP ALU op	Another FP ALU op	4
FP ALU op	Store double	3
Load double	FP ALU op	1
Load double	Store double	0

The execution cycles (including stalls) of one iteration can be shown as follows:

Instructions	Cycles
L.D	1
stall	2
ADD.D	3
stall	4
stall	5
stall	6
S.D	7
DADDUI	8
stall	9
BNE	10

7. Unroll the loops three times and apply instruction rescheduling so stall can be completely eliminated (assuming unlimited number of registers to use). Show the scheduling and cycle table similar to the above and calculate cycles per iteration of the original loop (to compute one element). (10 points)

Answer:

Instructions	Cycles
LD	1
LD	2
LD	3
ADD	4
ADD	5
ADD	6
DADDUI	7
SD	8
SD	9
SD	10
BNE	11

Cycles per iteration = 11/3 = 3.67

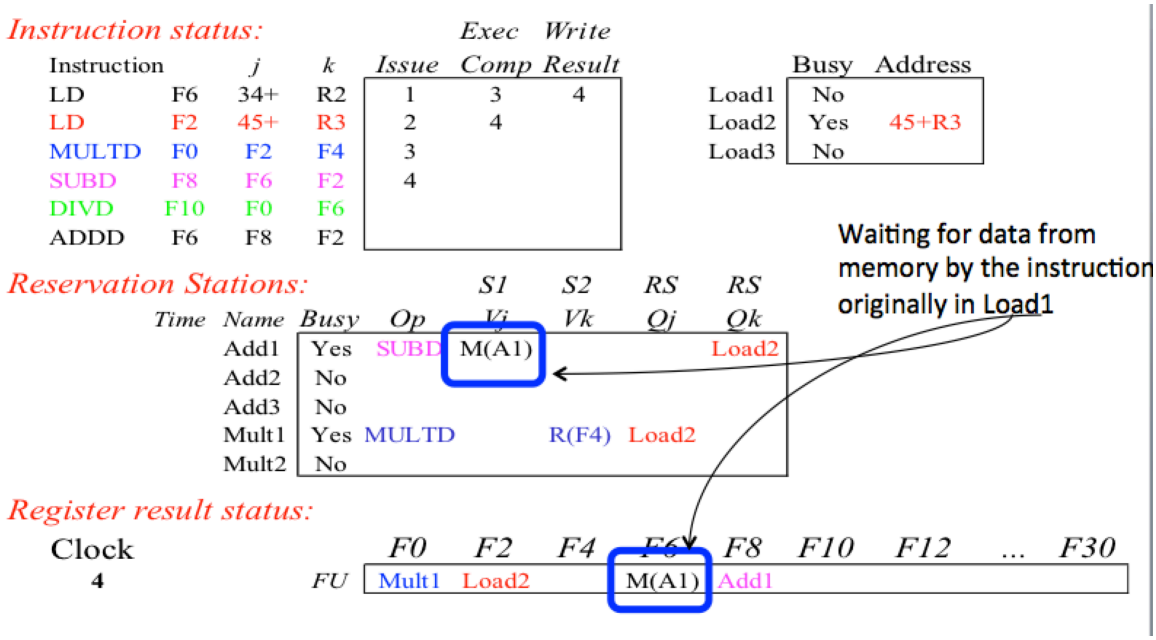
8. Unroll the loops 10 times and schedule instructions to achieve minimum number of cycles of execution in a VLIW machine of which each VLIW can include two LD/ST operations, two FP operations and one integer ALU/branch operations. Please include your solution in the following scheduling table (not necessarily 16 cycles). (10 points)

Cycles	LD/ST 1	LD/ST 2	FP 1	FP 2	Integer ALU/Branch
1					
2					

3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Cycles	LD/ST 1	LD/ST 2	FP 1	FP 2	Integer ALU/Branch
1	LD	LD			
2	LD	LD			
3	LD	LD	ADD	ADD	
4	LD	LD	ADD	ADD	
5	LD	LD	ADD	ADD	
6			ADD	ADD	
7	SD	SD	ADD	ADD	
8	SD	SD			
9	SD	SD			DADDUI
10	SD	SD			
11	SD	SD			BNE

9. Explain how register renaming and out-of-order execution are achieved in the Tomasolu algorithm using the following diagram. (10 points)



10. Explain how does the reorder-of-buffer work for branch prediction and hardware speculation. (6 points)

11. Explain the differences between Coarse-Grained Multithreading, Find-Grained Multithreading and Simultaneous Multithreading. (6 points)

12. For a machine with max vector length 64, the following code

```
int N = 64;
for (i=0; i<N; i++)
  Y[i] = a* X[i] + Y[i];
```

can be vectorized as:

```
L.D       F0,a        ;load scalar a
LV        V1,Rx       ;load vector X
MULVS.D   V2,V1,F0    ;vector-scalar multiply
LV        V3,Ry       ;load vector Y
ADDVV.D   V4,V2,V3    ;add
SV        V4,Ry       ;store the result
```

If N is 96, write the assembly code for vectorizing the loop using strip mining technique with vector length register. The instructions for read/write the vector length register are as follows: (10 points)

MTC1	VLR,R1	Move contents of R1 to vector-length register VL.
MFC1	R1,VLR	Move the contents of vector-length register VL to R1.

```
Add R1, R0, 96
SUB R1, R1, 64
MTC1 VLR, R1
<<the above sequence>>
ADD R1, R0, 64
MTC1 VLR, R1
<<the above sequence>>
```

13. For scheduling multiple threads in CPU, switching a thread on or off a CPU core involves restoring or storing the thread state (registers, PC, etc), which makes thread scheduling a costly operation. However, for GPU manycore thread scheduling, even threads are scheduled and executed in a warp group (32 threads), switching a warp of threads on and off GPU is very efficient. Explain what GPU feature makes this possible? (5 points)

14. Explain the cache coherence problem in MIMD architecture, e.g. symmetric multiple processor systems and explain how write-invalidate cache snooping protocol works to provide cache coherence for parallel programs. (8 points)

15. In this question, you need to describe the false sharing problem and propose a software solution and a hardware solution. (Total 15 points bonus)

    1). What is false sharing and how it is caused.

2). Using a program example to explain how to eliminate false sharing by padding.

3). From the architecture point view, please propose a solution in the cache architecture to eliminate false sharing.

---------- The end of the Test ----------

Full Name: _____ NetID: _____