Qualify Exam.: Computer Architecture July 2013

1. (40 %)

1.1 **(5%)** In a virtual memory system, the virtual address has 64 bits in which 13 bits are for page offset. Each entry in the page table has also 64 bits in which 30 bits are allocated to specify the physical page number. What is the largest amount of DRAM (primary memory) that can be usefully put in the virtual memory system?

1.2 **(5%)** Explain translation lookaside buffer (TLB) in a virtual memory system.

1.3 **(5%)** How many total bits are required for a direct-map cache with 16K bytes of data and 16-byte blocks, assuming a 32-bit address and one valid bit for each cache line?

1.4 **(5%)** Repeat the above problem for a two-way set-associative cache.

1.5 **(10%)** Compare the differences of three different cache designs: direct-mapped, set-associative, and fully-associative caches. What designs are usually used for level-one cache? What designs are usually used for TLB?

1.6 **(10%)** Give five methods to increase the cache performance. Explain what improvements (hit time, miss penalty, and/or miss rate) are achieved for each method.

2. (60%)

Consider the following C code:

```
void daxpy (double a,   double *x,   double *y,   int n) {
    for (int i=0; i<n; i++)
        y[i] = y[i] + a * x[i];
}
```

Assuming that the scalar **a** is stored in the 64-bit floating-point register **f4**, pointer **x** is stored in the 32-bit integer register **r5**, pointer **y** is store in the 32-bit integer register **r6**, and the loop range **n** is stored in the 32-bit integer register **r7**, the code compiles to the following inner loop:

```
loop:
I1:  fld    f1,  0(r5)     // x[i] = f1 = Mem[r5+0]
I2:  fld    f2,  0(r6)     // y[i] = f2 = Mem[r6+0]
I3:  fmul   f3,  f1,  f4   // a is stored in reg. f4
I4:  fadd   f1,  f2,  f3
I5:  fsd    f1,  0(r6)
I6:  addi   r5,  r5,  8
I7:  addi   r6,  r6,  8
I8:  addi   r7,  r7,  -1
I9:  bne    r7,  r0,  loop
```

2.1 (15%) Draw arrows to mark all register RAW, WAR, and WAW hazards between instructions in the above instruction sequence. How many RAW hazards are detected in one loop

iteration? How many WAR hazards are detected in one loop iteration? How many WAW hazards are detected in one loop iteration?

2.2 (5%) Continued with the previous problem, which of the data hazards (RAW, WAR, WAW) can be removed by renaming the registers? RAW, and/or WAR, and/or WAW?

2.3 (10%) The following tables show the latencies of floating-point (FP) operations for a particular CPU hardware. For example, if a FP ALU operation uses the results of another FP ALU operation, these two FP ALU instructions should be separated by at least 3 clock cycles. We assume the standard five-stage integer pipeline, so that branches have a delay of one clock cycle. In other words, a branch instruction should be separated at least one cycle from the instruction that computes the value of the register used by the branch. We also assume that the floating-point function units are fully pipelined so that an operation of any type can be issued on every clock cycle and there are no structural hazards. Show the instruction sequence (including stalls) for an *unscheduled* loop instruction in this CPU. How many clock cycles (including stall cycles) are required to issue the instruction sequence?

instruction producing result	instruction using result	latency in clock cycles
FP ALU operation	another FP ALU operation	3
FP ALU operation	store double	2
load double	FP ALU operation	1
load double	store double	0

2.4 (10%) Present an instruction scheduling method for the above instruction sequence to improve the instruction level parallelism.

2.5 (10%) If the C code is changed to the following (i.e., all variables are changed to integer format):

```
void daxpy (int a,   int *x,   int *y,   int n) {
    for (int i=0; i<n; i++)
        y[i] = y[i] + a * x[i];
}
```

Write the corresponding instruction sequences after compilation. Hint: replace the 64-bit floating registers by 32-bit fixed-point integer registers.

2.6 (10%) Explain dynamic scheduling with hardware speculation.