CSE 378 Machine Architecture and Assembly Language Wi10

PRACTICE QUESTIONS:

1) Programming in SPIM. Write a program in MIPS assembly language that finds (i) the number of elements strictly greater than the last element, (ii) the minimum element, (iii) the maximum element, and (iv) the (integer) average of all elements in an array of integers (each element is a 32-bit positive, negative, or null integer). Your input, the array and its size, should be in ".data" section of your program. As output, your program should display the 4 values asked for above on the console with appropriate messages.

2) Implement the following in MIPS:

(i) forloop.c - A for loop

```
int forloop(){
   int i = 0;
   int result = 0;
   for (i; i<20; i=i+2){
      result += i;
   }
   return result;
}
```

(ii) funccall.c - A function calling example

```
int main(){
   int a = 4;
   return helper(a);
}

int helper(int a){
   int b = a * 4;
   return b;
}
```

(iii) strcat.c - C string concatenation function

```
char* strcat (char * dst, char * src){
   char* originalDst = dst;
   while(*dst){
      dst++;
   }
   while(*src){
      *dst = *src;
      dst++;
      src++;
   }
   *dst = *src;
   return originalDst;
}
```

3) We wish to add the instruction addi (add immediate) to the single-cycle datapath described in this chapter. Add any necessary datapaths and control signals to the single-cycle datapath of (3rd Edition) Figure 5.17 on page 307 (4th Edition) Figure 4.17 on page 322 and show the necessary additions to (3rd Edition) Figure 5.18 on page 308 (4th Edition) Figure 4.19 on page 324. You can photocopy these figures to make it faster to show the additions.

4) This question is similar to the previous one except that we wish to add the instruction jal (jump and link).

5) This question is similar to the previous one except that we wish to add a variant of the lw (load word) instruction, which sums two registers to obtain the address of the data to be loaded and uses the R-format.

6) Identify all of the data dependencies in the following code. Which dependencies are data hazards that will be resolved via forwarding? Which dependencies are data hazards that will cause a stall?

```
add   $3, $4, $2
sub   $5, $3, $1
lw    $6, 200($3)
add   $7, $3, $6
```

7) Consider executing the following code on the pipelined data-path of (3rd Edition) Figure 6.36 on page 416 (4th Edition) Figure 4.60 on page 375:

```
lw    $4, 100($2)
sub   $6, $4, $3
add   $2, $3, $5
```

How many cycles will it take to execute this code? Draw a diagram like that of figure (3rd Edition) 6.34 on page 414 (4th Edition) Figure 4.58 on page 372 that illustrates the dependencies that need to be resolved, and provide another diagram like that of (3rd Edition) Figure 6.35 on page 415 (4th Edition) Figure 4.59 on page 374 (that illustrates how the code will actually be executed (incorporating any stalls or forwarding) so as to resolve the identified problems.