Name: _____ Points: _____

Last (family) name: _____

First (given) name: _____

Student I.D. #: _____

Department of Electrical and Computer Engineering
University of Wisconsin - Madison

ECE 902 Embedded Computing Systems

Midterm Exam - Solution

Tuesday, October 20, 2009
11:00AM--12:30PM ***(90 minutes)***

Instructions:

1. The exam is open book and open note.
2. You may find it useful to read over the entire exam before starting it.
3. Where appropriate, show your work.
4. No one shall leave the room during the last 5 minutes of the examination. If you need to leave the room before then, ask for permission.
5. Upon announcement of the end of the exam, stop writing and turn your exam face down. I will come by to pick them up.
6. Do not leave the room at any point unless you are finished with the exam or by permission of the instructor.
7. If you have any questions, please raise your hand.

For several problems, other answers may be correct.

Problem	*Points*	*Score*
1	30	
2	20	
3	28	
4	22	
Total	100	

Name: _____ Points: _____

[1] (30 points) Short answers.

a) (4 points) Explain why a single centralized register file typically becomes inefficient when a VLIW processor has a large number of functional units. What are two techniques for overcoming this inefficiency in VLIW processors?

A centralized register file needs a large number of read and write ports when there are a large number of functional units to prevent it from becoming a bottleneck in the system. However, the large number of read and write ports increases the register file area, delay and power.

Two techniques for overcoming the inefficiencies of large registers files are:
(1) Dividing the VLIW architecture into clusters, where each cluster has its own register file and set of functional units.
(2) Utilizing data buffers to store temporary results, as was done in AnySP.

b) (4 points) What are two types of security attacks that dynamic voltage and frequency scaling (DVFS) can help prevent? Explain how DVFS can be used to help prevent each type of attack.

DVFS can help prevent timing and power attacks. It can prevent timing attacks by adjusting the frequency in a pseudo-random manner to make it difficult to determine the data or key used for encryption simply by monitoring the time it takes for different encryptions. It can prevent power attacks by adjusting the frequency and/or supply voltage in a pseudo-random manner to make it difficult to determine the data or key used for encryption simply by monitoring the current used for different encryptions.

c) (3 points) What are three ways in which the instruction set architecture (ISA) of programmable digital signal processors typically differs from the ISA of general purpose processors?

Unlike the ISA of general purpose processors, the ISAs of digital signal processors typically:
(1) Use specialized addressing modes
(2) Have fewer general purpose registers
(3) Utilize VLIW architectures

d) (3 points) What are two instruction set architecture features used by processors to help reduce code size? For each feature, list a processor that we have studied in class that utilizes this feature.

Two instruction set architecture features used by processors to help reduce code size are
(1) Complex instructions – such as those utilized by the Sandblaster processor
(2) Vector instructions – such as those utilized by AnySP and Sandblaster

e) (4 points) What are two characteristics of video processing algorithms that the AnySP processor is designed to handle efficiently? For each characteristic, explain how the AnySP processor is designed to handle the characteristic?

Two characteristics of video processing algorithms that the AnySP processor is designed to handle are:
(1) Instruction pairs, such as multiply-add and shift-add are frequently executed. To handle this, AnySP uses flexible functional units that allow pairs of instructions to be merged.
(2) Many variables have short lifetimes. To take advantage of this, AnySP utilizes buffers that store temporary variables with short lifetimes so that these variables do not need to be read from and written to the register file.

Name: _____ Points: _____

f) (4 points) How does the Sandblaster Processor eliminate the need for dependency checking and bypass hardware? What are two disadvantages of this approach?

The Sandblaster processor uses T^3 multithreading to ensure that an instruction does not read its results before the previous instruction from the same thread has written the result back from the register file. Two disadvantages of this approach are (1) it limits single thread performance, and (2) it leaves datapath resources underutilized when there are not enough threads.

g) (4 points) Explain why configurable processors can provide better energy efficiency than general purpose embedded processors. Give two disadvantages that configurable processor have compared to general purpose embedded processors.

Configurable processors can provide better energy efficiency than general purpose embedded processors because their hardware resources (e.g., register files, caches, functional unit available, etc.) and instruction set can be tailored to the requirements of the target applications, thus eliminating the overhead of unnecessary resources and instructions. Two disadvantage that configurable processors have compared to general purpose embedded processors include (1) they have poor binary compatibility and (2) they are less likely to perform well on a wide range of embedded applications.

h) (4 points) When applying the dictionary-based bus encoding scheme to an address bus, how could you determine suitable widths for the upper, index, and offset portions of the address? Give an equation that shows how the total size of the dictionary (S), in bits can be computed using the number of bits in the upper (N_u), index (N_i), and offset (N_o) portions.

When applying the dictionary-based bus encoding scheme to an address bus, suitable widths for the upper, index, and offset portions of the address can be obtained by profiling the code to determine the addresses transmitted across the bus. The widths of the upper and index bits should be selected such that a large number of addresses have the same upper and index bits in common since this will lead to more hits in the dictionary, while the offset portion should correspond to the bits of the address that change frequently. When selecting the width of the index bits, it is also important to make sure that not too many index bits are used, since the size of the table grows exponentially with the number of index bits.

The table size can be computed as $S = N_u \times 2^{N_i}$.

[2] (20 points) Code and Data Compression.

a) (8 points) Fill in the rest of the table below for LZW compression. Assume the basic elements correspond to the 16 4-bit hexadecimal digits 0 through F and that these elements are originally stored in the table at locations 0 through 15. Also assume that the table can hold a total of 32 16-bit values. The "Output" should be given in decimal and the "Added to Table" column should include the hexadecimal value that was added to the table during each time step followed by (in parenthesis) the location in the table where the value was added (in decimal). Thus, 0F (16) indicates that the hexadecimal value 0F is stored in the table at location 16. Use a "–" to indicate that no value is output or added to the table. The input string in hexadecimal is 0FAF0FAF0F.

Time	Input (hexadecimal)	Output (decimal)	Added to Table

Name: _____ Points: _____

0	-	-	0 (0), 1 (1), …, F(15)
1	0	-	-
2	F	0	0F (16)
3	A	15	FA (17)
4	F	**10**	**AF (18)**
5	0	**15**	**F0 (19)**
6	F	**-**	**-**
7	A	**16**	**0FA (20)**
8	F	**-**	**-**
9	0	**18**	**AF0 (21)**
10	F	**-**	**-**
11	-	**16**	**-**

b) (2 points) What compression ratio is achieved for the compression described in problem (2a)? Show your work.

Since the table has 32 entries, each symbol in the compressed code should be 5 bits. A total of 7 symbols were needed to transmit the 10 4-bit hexadecimal sequences 0FAF0FAF0f. With LZW, the table is generated on the fly and should not be counted as part of the compressed code size.

Compression ratio = compressed code/uncompressed code = (7 x 5)/(10 x 4) = 35/40 = 87.5%

c) (6 points) If you use Huffman encoding to encode 0FAF0FAF0F, what is the compression ratio? Assume each input symbol is one byte.

With Huffman coding and one byte input symbols, the only input symbols are 0Fand AF. With Huffman encoding, 0F can be encoded as '1' and AF can be encoded as '0'. Thus transmitting five byes only requires five bits. However, we also need to include the table, which stores AF at location 0 and AF at location 1. This requires a 2-byte table.

Compression ratio = compressed code/uncompressed code = (5 + 2 x 8)/(10 x 4) = 21/40 = 52.5%

d) (4 points) Explain how branch patching works. What are one advantage and one disadvantage that branch patching has compared to using branch tables?

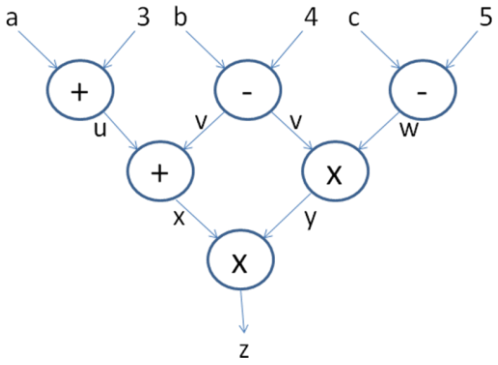
With branch patching, branch instructions are compressed using a special format and a second pass through the compressed code is utilized to patch the branch instructions so that they point to addresses in the compressed code, rather than addresses in the uncompressed code. Compared to using branch tables, branch patching avoids the area, power, and delay needed to store and access the branch table. However, the compression ratio with branch patching may not be as high, since branch instructions are not compressed as much.

Name: _____ Points: _____

[3] (28 points) Models of Computation and VLIW Scheduling

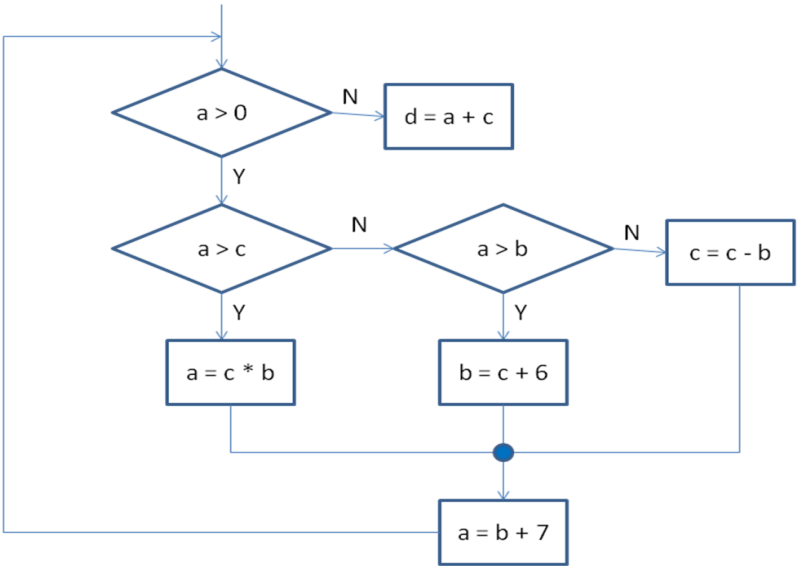
a) (6 points) Draw a dataflow graph for the following computations. Label each variable.

```
u = a + 3;
v = b – 4;
w = c – 5;
x = u + v;
y = w * v;
z = x * y;
```



b) (8 points) Write a control data flow graph for the following code.

```
while (a > 0) {
        if (a > c)
            a = c*b;
        else if (a > b)
            b = c + 6;
         else
            c = c – b;
         a = b +7;
}
d = a + c;
```



Name: _____ Points: _____

c) (4 points) If you needed to pick one of the processors that we studied in class to very quickly implement the above code, which one would you chose? Assume that all variables are 32-bit integers in registers and that a, b, and c are not needed once d is computed. Justify your answer.

One good choice for implementing the above code would be the Xtensa processor, since the loop body could be implemented as a specialized instruction.

d) (10 points) A VLIW processor has the instruction format given below.

Branch	Arithmetic	Arithmetic	Arithmetic	Memory	Memory

Assume the "Branch" slot can be used for branches or jumps, the "Arithmetic" slot can be used for any arithmetic operation, and the "Memory" slot can be used for loads and stores. As an example, the VLIW instruction shown below implments a three arithmetic operations (add, subtact, and multiply), a load (into r2 from the memory location 10), and a store (into memory location 20 with the data from r8).

nop	add r1 = r2,r3	sub r4 = r5,r6	mul r7= r8,r9	lw r2=m[10]	sw m[20]= r8

Schedule the following operations using the four VLIWs given below such that the time needed to execute the code is minimized. Be sure to show any slots that should be filled with no-ops. You may find it useful to draw a data flow graph of the computations. Assume that lw and mul have a latency of 2 cycles and all other instructions have a latency of one cycle.

```
lw    r1 = m[100];
lw    r2 = m[200];
lw    r3 = m[300];
add   r4 = r1 * r2;
sub   r5 = r0, #6;
sub   r6 = r0, #3;
mul   r7 = r4, r3;
mul   r8 = r1, r5
mul   r9 = r2, r5
add   r10 = r3, r2;
sw    m[35] = r4;
```

To determine which operations go in each VLIW it is useful to determine data dependencies between the various instructions and make sure that each instruction has all the required inputs by the cycle in which it is scheduled.

nop	**sub r5**	**sub r6**	**nop**	**lw r1**	**lw r2**

nop	**nop**	**nop**	**nop**	**nop**	**lw r3**

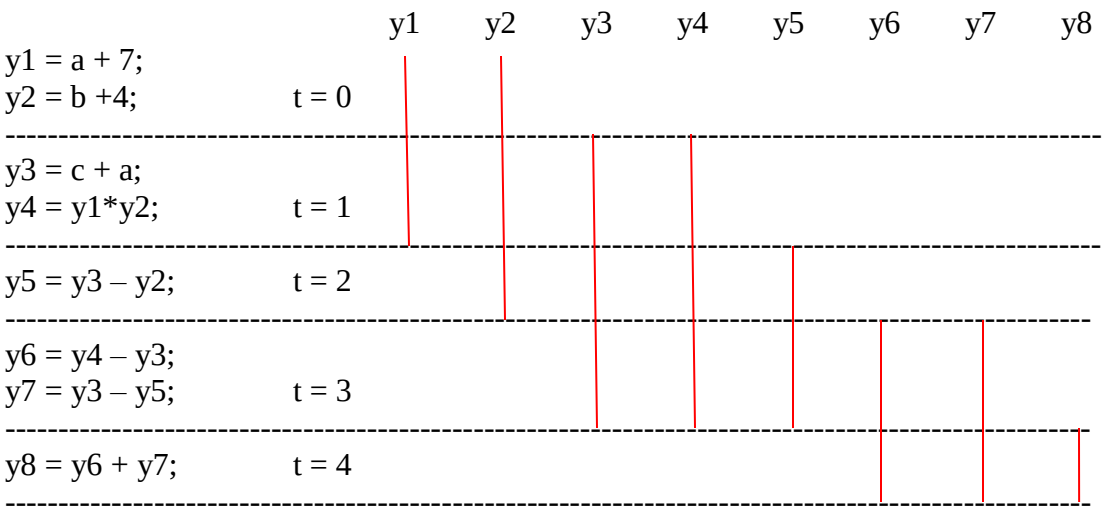
nop	**add r4**	**mul r8**	**mul r9**	**nop**	**nop**

nop	**mul r7**	**add r10**	**nop**	**nop**	**sw m[35]**

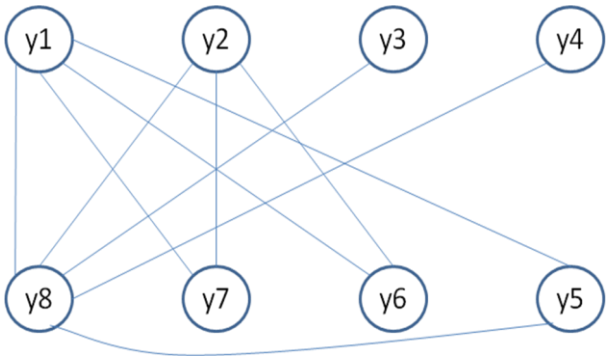
Name: _____ Points: _____

[4] (22 points) Code Generation and Back-end Compilation

a) (8 points) Complete the variable lifetime analysis chart given below. Assume that the program completes at the end of the code.



b) (8 points) For the lifetime analysis chart given in part (4a), draw the conflict graph and provide a register assignment that only uses five registers (r1 through r5) for variables y1 through y8. Assume that a variable cannot be both read and written in the same cycle.



The conflict graph connects variables with non-overlapping lifetimes. Variables with non-overlapping lifetimes can be assigned to the same register. One possible register assignment is: r1 gets y1 and y7, r2 gets y2 and y6, r3 gets y3 and y8, r4 gets y4, r5 gets y5.

c) (6 points) Explain how placement of subroutines can be used to improve energy and performance in embedded systems. How does one determine which subroutines to move and where to move them?

To improve energy and performance, subroutines should be placed such that they will not have cache conflicts with the routines that call them or with other routines that are frequently accessed close in time. To determine which routines to move and where to move them, the code can be profiled with representative inputs. A call graph can then be constructed to identify which routines frequently call other routines and which routines have cache conflicts. The profiling information can then be used to move routines that frequently conflict to other locations so that they no longer conflict. By reducing conflict misses in the instruction cache the energy and performance of the system is improved.

Name: ________________________________ Points: _________