CSE 320 Honors Project
Milestone #2

Milestone Deliverables

Use the "handin" system to submit the following files:

- mathunit.h
- mathunit.c

I recommend completing the milestone by Tuesday, February 16 (it must be completed no later than Tuesday, March 1).

Please note that the "handin" system is expecting you to use "320H" as your course and "9" as your section.

Milestone Overview

The ARC+ microprocessor contains circuits to perform a variety of operations on 32-bit integer data, including arithmetic, bitwise and shift operations.

This milestone focuses on the circuits which perform the operations listed in the table below.

Operation	Opcode	Description
Add	000	Addition (A + B)
AND	001	Bitwise AND (A AND B)
OR	010	Bitwise OR (A OR B)
XOR	011	Bitwise XOR (A XOR B)
Sub	100	Subtraction (A + ~B + 1)
ANDN	101	Bitwise ANDN (A AND ~B)
ORN	110	Bitwise ORN (A OR ~B)
XNOR	111	Bitwise XNOR (A XOR ~B)

For the four operations in the set {Sub, ANDN, ORN, XNOR}, the second operand is inverted (each bit in the 32-bit operand is complemented).

All eight operations produce a 4-bit condition code, as defined for the SPARC "cc" instructions on pages 172-174 of the SPARC Architecture Manual.

For our convenience, we'll refer to these circuits as the Math Unit. In a subsequent milestone, we'll develop the Shift Unit.

Milestone Resources

The Murdocca and Heuring textbook contains information about several topics which are relevant for this milestone:

- Twos complement representation of signed integers (pages 31-32)
- Addition and subtraction (pages 61-67, particularly Figure 3-6)

The SPARC Architecture Manual specifies the behavior of the operations:

- Addition (page 108 and page 173)
- Subtraction (page 110 and page 174)
- Bitwise operations (page 106 and page 172)

For the time being, you'll ignore all details related to the format of machine language instructions (such as registers and the "simm13" field).

Milestone Preparation

I have provided a framework for you to experiment with an 8-bit version of the Math Unit. The following files are available on the system under "/user/cse320/Honors/Milestone02":

```
math8.h         -- interface for the 8-bit Math Unit
math8.c         -- implementation of the 8-bit Math Unit (incomplete)
math8.test.c    -- test bed for the 8-bit Math Unit
math8.test.make -- makefile for the test bed
```

I recommend the following steps:

a) Create a directory for this milestone and copy the files (above).

b) Examine "math8.test.make" to familiarize yourself with that makefile.

c) Use the makefile to create an executable version of the test bed and the 8-bit Math Unit (which is currently incomplete), then run the test bed and observe the behavior. Note that the 8-bit output signal is uninitialized for the AND and OR operations.

d) Modify the 8-bit Math Unit to correctly process the missing operations (AND, OR), then test your work.

e) Modify the 8-bit Math Unit to correctly invert the second operand (using XOR gates), then test your work.

f) Modify the 8-bit Math Unit to correctly generate the 4-bit condition codes, then test your work.

Milestone Specifications

Copy your completed versions of "math8.h" and "math8.c" to create the two deliverables for this milestone ("mathunit.h" and "mathunit.c"), then edit those files to implement a 32-bit Math Unit.

Verify that your 32-bit Math Unit is correct using the following files:

```
mathunit.check.o
mathunit.check.make
```

Note that "mathunit.check" displays the following items:

```
Opcode (3 bits)
Operand #1 (32 bits)
Operand #2 (32 bits)
Result (32 bits)
Condition codes (4 bits)
```

All values are displayed in hexadecimal and are written to the standard error stream.