CSE 320 Spring 2013

Computer Project #7 -- The SPARC Instruction Set Architecture

Assignment Overview:

This assignment develops familiarity with the instruction set architecture and machine language of the SPARC microprocessor.

It is worth 40 points (4% of course grade), and must be completed no later than 11:59 PM on Thursday, March 21.

Assignment Specifications:

1) The program will simulate the actions of the CPU during the instruction fetch-execute cycle for the SPARC microprocessor.

2) The instructor-supplied driver module, which manages the instruction cycle as a whole, contains the definition of function "main" and some auxiliary functions. For each machine language instruction to be processed, it will simulate the steps of the fetch phase of the cycle and will call function "execute" to simulate the steps of the execute phase.

You will develop function "execute", which will decode and process the instruction in the IR, then update the register file and the PSR (if appropriate).

The declarations for the instructor-supplied data objects and functions, as well as function "execute", are appended below.

3) Assume that the microprocessor only recognizes a subset of the full SPARC instruction set. The legal instructions are listed below.

Arithmetic:	ADD, SUB, ADDX, SUBX
Logical:	AND, OR, XOR, ANDN, ORN, XNOR
Shifting:	SLL, SRL, SRA
Other:	SETHI

The microprocessor does not recognize the "CC" versions of the arithmetic and logical instructions ("ADDcc", for example).

4) The PSR register contains the condition code bits (in bits 23-20) and the following exception bit.

Bit 0:   Illegal instruction exception

Function "execute" will set the exception bit, as appropriate for the current machine language instruction.

Assignment Deliverables:

The deliverables for this assignment are:

```
proj07.makefile  -- the makefile which produces "proj07"
proj07.support.c -- the source code for your support module
```

Be sure to submit your files for grading via the "handin" program.

Contents of "/user/cse320/Projects/project07.hardware.h"

```
/******************************************************************************
  Instructor-supplied data types, data objects, and functions
******************************************************************************/

typedef unsigned signal1;
typedef unsigned signal2;
typedef unsigned signal3;
typedef unsigned signal5;
typedef unsigned signal6;
typedef unsigned signal13;
typedef unsigned signal22;
typedef unsigned signal32;

/*-----------------------------------------------------------------------------
  The control registers.
-----------------------------------------------------------------------------*/

extern signal32 IR, PSR;

/*-----------------------------------------------------------------------------
  Function:  read_reg_file

  Purpose:   Fetch the contents of two registers from the register file.
    RS1 output port <== Reg[RS1 selection signal]
    RS2 output port <== Reg[RS2 selection signal]
-----------------------------------------------------------------------------*/

void read_reg_file
(
  signal5,    /* RS1 selection signal */
  signal5,    /* RS2 selection signal */
  signal32*,  /* RS1 output port      */
  signal32*   /* RS2 output port      */
);

/*-----------------------------------------------------------------------------
  Function:  write_reg_file

  Purpose:   Update the contents of one register in the register file.
    Reg[RD selection signal] <== RD input port
-----------------------------------------------------------------------------*/

void write_reg_file
(
  signal5,    /* RD selection signal  */
  signal32    /* RD input port        */
);
```

Contents of "/user/cse320/Projects/project07.support.h"

```
/******************************************************************************
  Student-supplied function
******************************************************************************/

void execute();
```