RISC Processor Simulator (SRC)
INEL 4215: Computer Architecture and Organization
Feb 14, 2005

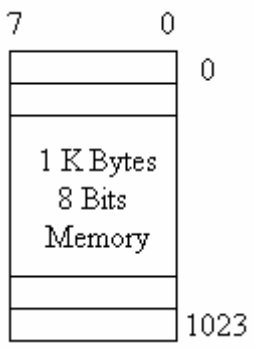
General Project Description

In the textbook, "Computer Systems Design and Architecture" by Heuring, we have the description of a RISC processor called the Simple Risc Computer (SRC). This processor is used as an example through the book to show all the different aspects of an architecture and all the components and parts of processors.

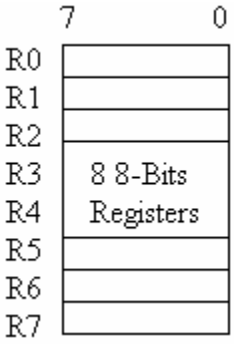
In order to undertand how this processor works and the different aspects taken into consideration when designing a processor, we are requesting that you design a simulator for a simple version of the SRC processor. Students will form groups of 3 or 4 students to develop your own version of the simulator. The following sections describe the processor.

Memory and Registers

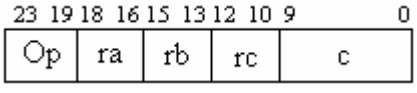
The size of the memory is 1KB organizad as 1024 addresses of 1 byte each.



Internally, the processor has 8 general purpose registers, 8 bits each. The names of the registers are from **r0** to **r7**.



The processor has a 10 bit program counter called **PC** and a 24 bit instruction register called **IR**. The following figure illustrates the **IR** format:



Op : 5 bits opcode. Determines which instruction will be executed.

ra: 3 bits which indicate which one of the eight registers will be register **a**
rb: 3 bits which indicate which one of the eight registers will be register **b**
rc: 3 bits which indicate which one of the eight registers will be register **c**
c: 10 bits representing **c1**, **c2** y **c3** from the SRC illustrated in the textbook.

Instruction Set
You will find a description of the instruction set of the processor in sections 2.3 to 2.5 in the textbook and appendix B. The following table is a summary of the instruction set. Instruction MAC follows the format of instruction ADD.

op	Name	Operands	Operation	Details
0	NOP		Does nothing	
1	LD	RA,RB,C	If rb = 0 R[ra] = M[c] Else R[ra] = M[c+R[rb]]	
2	LDR	RA,C	R[ra] = M[pc + c]	
3	ST	RA,RB,C	If rb = 0 M[c2] = R[ra] Else M[c2+R[rb]] = R[ra]	
4	STR	RA,C	M[pc+c] = R[ra]	
5	LA	RA,RB,C	If rb = 0 R[ra] = c Else R[ra] = c + R[rb]	
6	LAR	RA,C	R[ra] = PC + c	
8	BR	RB,RC,cond	If cond is true PC = R[rb]	Conditions 000 don't jump 001 jump always 010 R[rc] = 0 011 R[rc] <> 0 100 R[rc] >= 0 101 R[rc] < 0
9	BRL	RA,RB,RC,cond	R[ra] = pc If cond is true PC = R[rb]	Conditions are the same as above
10	MAC	RA,RB,RC	R[ra] = R[ra] * R[rb] + R[rc]	Add group of three registers Only four least significant bits Of ra and rb
12	ADD	RA,RB,RC	R[ra] = R[rb] + R[rc]	
13	ADDI	RA,RB,C	R[ra] = R[rb] + c	
14	SUB	RA,RB,RC	R[ra] = R[rb] – R[rc]	
15	NEG	RA,RC	R[ra] = -R[rc]	
20	AND	RA,RB,RC	R[ra] = R[rb] ^ R[rc]	
21	ANDI	RA,RB,C	R[ra] = R[rb] ^ c	
22	OR	RA,RB,RC	R[ra] = R[rb] v R[rc]	
23	ORI	RA,RB,C	R[ra] = R[rb] v c	
24	NOT	RA,RC	R[ra] = $\overline{R[rc]}$	
26	SHR	RA,RB,RC,count	If count = 0 R[ra] = R[rb] shr R[rc] Else R[ra] = R[rb] shr count	Only 5 lsb of RC are considered
27	SHRA	RA,RB,RC,count	If count = 0	Only 5 lsb of RC

			R[ra] = R[rb] shra R[rc] Else R[ra] = R[rb] shra count	are considered
28	SHL	RA,RB,RC,count	If count = 0 R[ra] = R[rb] shl R[rc] Else R[ra] = R[rb] shl count	Only 5 lsb of RC are considered
29	SHC	RA,RB,RC,count	If count = 0 R[ra] = R[rb] shc R[rc] Else R[ra] = R[rb] shc count	Only 5 lsb of RC are considered
31	STOP		Stop flag is set	

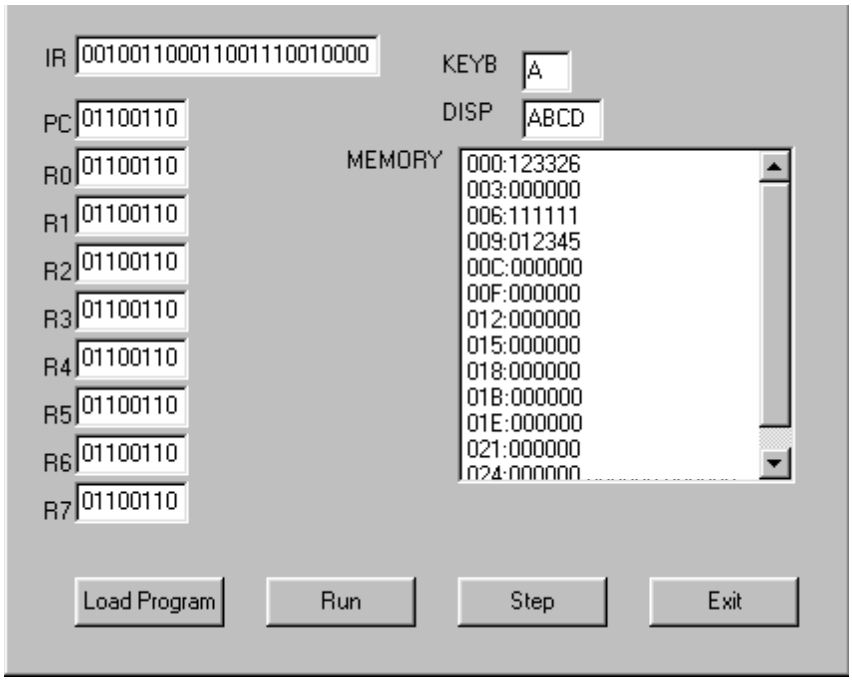
Arithmetic instructions use a 2's complement representation for negative numbers. This format is also used to compute memory addresses when accessing memory.

Processor Configuration

The processor operates with instructions located in main memory from address 0 to address 511. When the processor starts, it will always do so from location 0, 1 and 2.

The system will also have two devices connected to I/O ports using the following memory locations:
512-513: 16 bits, input from keyboard
514 – 517: Hex display, each byte will represent one digit

The information coming from the keyboard will be entered at the keyboard of the computer running the simulation. The hex display will be presented at the computer screen of the computer running the simulation. The following figure illustrates a possible configuration of the graphical user interface for the simulator you will design.



Due date

March 11, 2005

Project requirements

Three or four of students per group. Divide tasks as balanced as possible, and please take into consideration preferences and skills. As part of the report, I want to see a list of responsibilities and who completed them.

Simulator characteristics

The simulator must simulate all instructions in the instruction set, including all addressing modes. The simulator will run the instructions located in the main memory, starting with address zero (PC=0) The contents of memory are changed as a file with instructions is loaded into the simulator. Nayda Santiago will bring a simulation file on the oral exam day. This file will contain one line per instruction, and it will be represented in HEX characters (6 Hex characters).

The following example illustrates an input file:
2B001E
2C0004
1B8204
627000

The simulator should run in two different modes: run or step. Run mode will allow programs to run from start to end. Step mode will run one instruction at a time. Please notice that the last instruction in any file should be stop. The simulator must show the contents of all registers, program counter, instruction register, and the contents of a section of memory. The directions in memory should be shown in HEX representation

Your design may use a graphical user interface or a plain text interface. Use any programming language of your preference.

Project Report and Evaluation

Renglones ..... Puntos
Funcionamiento de las instrucciones ..... 20 puntos
Implementación de las operaciones aritméticas usando “2’s complement” ..... 5 puntos
Implementación de la funcion ADC ..... 5 puntos
Interfaz cumple con los requisitos ..... 10 puntos
El programa lee a archivos (I/O) como indicado ..... 10 puntos
Happy Hour ..... 15 puntos
    Puntaje individual para cada miembro del grupo.
Código fuente ..... 5 puntos
    Documentación del código es apropiada. Autor, día, descripción, variables, etc.
Reporte ..... 30 puntos
    Contiene todas sus partes:
        Cubierta (Binding, cartapacio, etc., debe verse bien)
        Página de título
        Resumen o “abstract”
        Tabla de contenido
        Lista de figuras
        Lista de tablas
        Introducción
        Discusión
        Conclusión
        Referencias
        Apéndice
        Un sobre con un CD conteniendo el código fuente del simulador.

Favor de referirse a *Writing Formal Reports: An Approach for Engineering Students in 21st Century*, Third Edition, by Nayda G. Santiago & Manuel A. Jiménez en la página web de la clase.

Introducción apropiada
Describe el diseño del código.
Incluye código fuente.
Concluye apropiadamente.
Distribución equitativa de tareas.
Incluye gant chart.
Analizaron el problema.
La explicación es apropiada.
Bien organizado y coherente.
Lenguaje apropiado.
Incluye las cartas en el apéndice

Penalties

Dia de entrega: -0 puntos.
Next day: -10 puntos
Two days late: -25 puntos
Three days late: -40 puntos
Four days late: not accepted