

Computer Architecture Laboratory

DRC Computer and Assembly Language

Goals:

- a) Become familiar with the DRC Simulated Computer
- b) To redesign the DRC to create an HRC
- c) To improve your technical writing skills by writing a report on the design of part b)

- 1) Read the module on the DRC Computer and its assembly language. This module may be accessed from our web page.
- 2) Try several examples on the DRC2 Simulator. A few examples are described below. Others are in the module. The instructor will show you how to access and use the simulator. Pay careful attention to the functioning of the simulator.
- 3) Draw a finite state diagram of how this simulator functions at the macroscopic level, from the editor to assembled program to the execution stage (including the run and single step modes). Use PowerPoint or a similar editor.
- 4) Redesign the DRC to become HRC (Hex RISC Computer). Change the machine from being decimal based to hex based. Pay attention to the following questions (among others):
 - What changes will take place to the registers and register file?
 - How much memory could you address?
 - What data values are possible?
 - What formats can instructions have?
 - How will you handle negative numbers?
 - What changes would be implied for the instruction set? Will you have 2 character opcodes (possible 256 instructions) or 1 character opcodes (only 16 instructions)? If the former, what instructions would you add? Why? If the latter, which instructions would you eliminate? Would you change any instructions? It is not necessary to "improve" the machine in this redesign, simply make it hex based.
- 5) Write a paper describing the current simulator and your new design. Be sure the paper addresses all the questions posed in part 4. It should give a complete specification for all parts of the datapath and instruction set.
 - The paper should combine text, assembly language statements, RTN, and diagrams into a smoothly blended whole.
 - For assistance, review Chapters 1 - 4 of [Writing for Computer Science](#) by Justin Zobel for suggestions on style and approach to the writing of this paper. Chapters 5 and 6 offer advice on presenting the instruction set and incorporating diagrams.

The paper is due in two weeks on December 2. DRC2 is available from the instructor.

Examples:

1. DRC does not have a multiply instruction. Write a program that brings in two numbers from memory and multiplies them by successive addition (i.e. 5×4 is 4 added to itself five times $4+4+4+4+4$) and stores the result back in memory. Your program should check that the product does not overflow the register size (If it does the value -9999 should be stored). You may assume positive numbers.
2. DRC does not have a division instruction either. Write a program that does short division storing both the result and remainder in successive memory locations. Division may be done by successive subtractions.
3. Write a program that examines a series of values in memory, selects the largest and stores it in at the end of the list. Store the number of values in location 70 and the values in 71..? depending on the number of values. Your loop should read the number of values from location 70 and should store the result (maximum value) immediately after the list of numbers. None of the data values should be changed.