Compter Architecture: My vision for the future

Yale Patt

The University of Texas at Austin

Workshop, UT Austin

September 19, 2014

What I want to do today

- ***First an admission***
- ***Then some observations***
- ***What do we do about it***
- ***How do we do it***
- ***What will be the (hopeful) result***

What I care about;
What I don't care about

- ***What I care about:***
  - ***Curing cancer, predicting tsunamis***
  - ***Harnessing the chip's resources to do great things***

- ***What I don't care about:***
  - ***Improving the performance of Microsoft Word***
  - ***Making it easier for some people to write payroll programs***

Problem

Algorithm

Program

ISA (Instruction Set Arch)

Microarchitecture

Circuits

Electrons

Observations

- ***GPUs are being pushed to do non-GPU stuff***
  - ***Scientific appllications, data base***
- ***Derek Chiou (with Microsoft's blessings)***
  - ***An FPGA fabric***
- ***Embedded controllers – domain specific***
  - ***Tuning from top to bottom***
- ***Accelerators***
  - ***Asynch for several cycles, then synch***
  - ***What do we accelerate***
- ***Moore's Law may not last forever***

More observations

- ***Latency as well as throughput***
  - ***Where does the run time system go?***
- ***Dark silicon***
  - ***A bug or a feature?***
- ***The silliness of a multiple-ISA, multi-core chip***
  - ***Endianness, page size, data types***
  - ***The good sense of a single-ISA, multiple-uarch cores***
- ***What % of the market cares about all of this***
  - ***I no longer have to embarrassedly say less than 1%***

Still more observations (these of a very different ilk)

- ***Best students don't choose computing any more***
  - ***Forget the favorite, tired cop-out: off-shoring***
- ***High school girls don't opt for engineering***
  - ***Is there a reason we have not explored***
- ***Freshman arrive knowing nothing re: computing***
  - ***Even with their AP 5 credential earned in high school***

What do we do about it?
We must break the layers

- ***(We already have in limited cases)***
- ***Pragmas in the Language***
- ***The Refrigerator***
- ***X + Superscalar***
- ***The algorithm, the language, the compiler, & the microarchitecture all working together***

IF we break the layers:

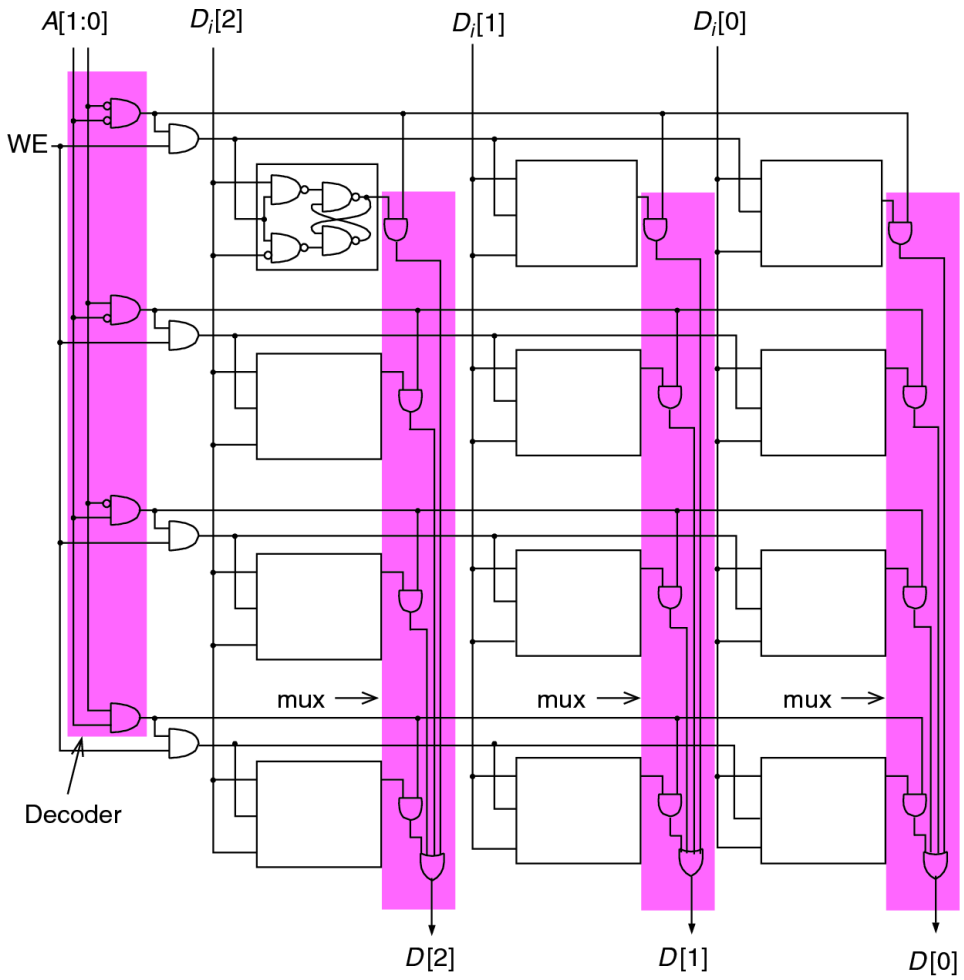
- ***Compiler, Microarchitecture***
  - ***Multiple levels of cache***
  - ***Block-structured ISA***
  - ***Part by compiler, part by uarch***
  - ***Fast track, slow track***

- ***Algorithm, Compiler, Microarchitecture***
  - ***X + superscalar – the Refrigerator***
  - ***Niagara X / Pentium Y***

- ***Microarchitecture, Circuits***
  - ***Verification Hooks***
  - ***Internal fault tolerance***

How do we do it?
We start in the freshman year

- ***Start with what they "know"***
  - ***The transistor as light switch***
  - ***Not quantum mechanics***
- ***Choose a computer model that is simple***
  - ***As the genius said: simple, but still rich***
- ***Continually build on what they know***
- ***Continually raising the level of abstraction***
- ***Memorizing as little as absolutely necessary***
- ***Trying very hard to not introduce magic***


A[1:0]
Di[2]
Di[1]
Di[0]
WE
mux
mux
mux
Decoder
D[2]
D[1]
D[0]

The ISA

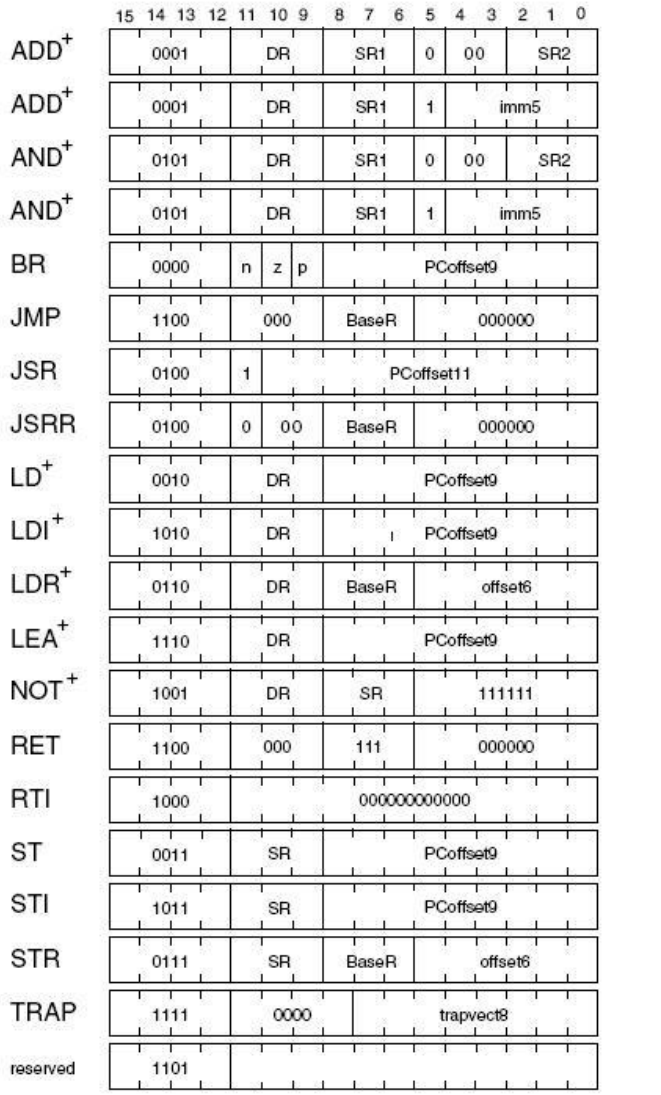


Figure A.2 Format of the entire LC-3 instruction set. Note: + indicates instructions that modify condition codes

The Data Path

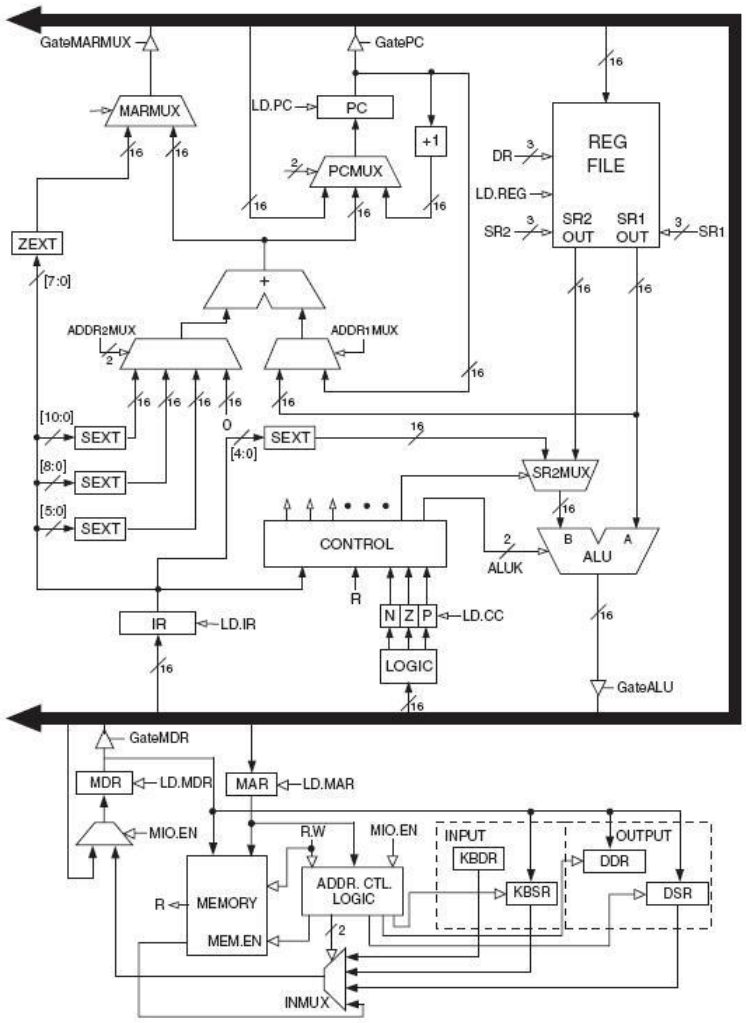


Figure C.3 The LC-3 data path

The State Machine

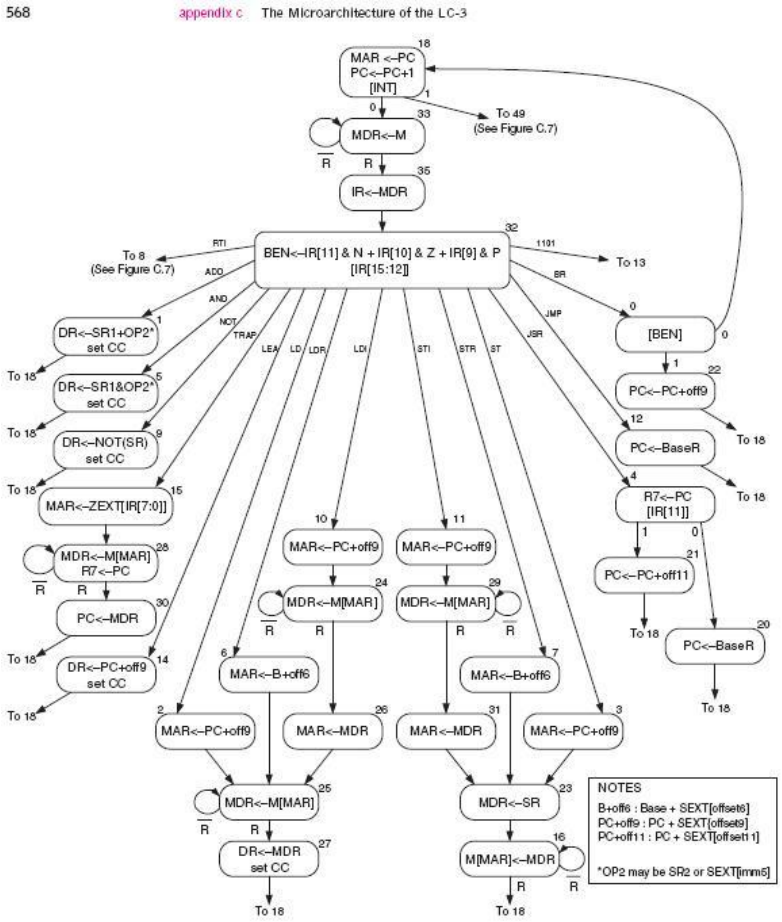


Figure C.2 A state machine for the LC-3

What I have learned about students

- ***Freshmen can handle serious meat***
- ***Students don't need glitz***
- ***Computer architecture can begin with freshmen***
- ***Students will debug their own programs***
  - ***Bottom-up eliminates memorization***
  - ***Memorizing 95% correctly gets you nowhere***
- ***Good students don't want to memorize***

What I know about education

- ***Engineering education (No substitute for):***
  - ***Design it wrong***
  - ***Debug it yourself***
  - ***Fix it***
  - ***See the working result***
- ***Avoid the latest fads***
  - ***Today the pressure is for freshmen to embrace JAVA***
  - ***By the time they graduate, C# will have replaced JAVA***
  - ***C# will probably be replaced by D-flat***
- ***Abstractions are great***
  - ***AFTER you understand what you are abstracting***

Most importantly

- ***This freshman course could be taught in high school***
- ***Nothing in it is beyond the ability of good students***
- ***The best and brightest boys and girls would sign up***
- ***They want something they can sink their teeth into***
- ***We would attract the boys and girls who will make good engineers***

- ***To get there we must do two things:***
  - ***Develop trained teachers***
  - ***Get NSF and ETS beyond the JAVA nonsense***

The (hoped for) result

- ***What I want***
  - ***The future of Computer Architecture can contribute to curing cancer, predicting tsunamis***
- ***What we need***
  - ***The best and brightest boys and girls coming out of K-12***
  - ***An education that gets them ready for these challenges***

Thank you!