

Topic 3 : Controlling the Computer

More Machine Language & Assembly Language
Reference : D & L pp 215- 222 but note that these pages are very much background reading.

C.Sc. 110: Computer Systems

Addressing Modes

- In previous examples the instructions have had the address of their operand hard-wired.
 - Load 5 : 1005
 - Store 5 : 2005
- Such a restriction makes it difficult to perform routine tasks such as:-
 - dealing with constants.
 - processing arrays.
 - dealing with complex data structures.

C.Sc. 110: Computer Systems

Addressing Modes

- Solution is to provide support for common addressing requirements by introducing more powerful addressing modes.
 - Direct
 - Immediate
 - Indexed
 - Indirect

C.Sc. 110: Computer Systems

Direct Addressing

LOAD 10

- Copies the value stored at the specified address into the accumulator.
- Used for accessing stored data.

C.Sc. 110: Computer Systems

Immediate Addressing

LOAD #10

- Copies the operand into the accumulator.
- Used for dealing with constants.

C.Sc. 110: Computer Systems

Indexed Addressing

LOAD +10

- Introduces a new register called the index register.
- Effective address = the operand field + the value in the index register.
- Used for processing arrays

C.Sc. 110: Computer Systems

Indirect Addressing

LOAD @10

- The operand specifies the address of the address of the data.
- Used for handling complex data structures.

C.Sc. 110: Computer Systems

Examples of Addressing Modes

ACC : 0

Load Immediate 110	110	210
	111	212
	112	7
	113	8
	114	9
	
	210	10
	211	11
	212	12
	213	13
	214	14

C.Sc. 110: Computer Systems Index 2

Examples of Addressing Modes

ACC : 110

Load Immediate 110	110	210
	111	212
	112	7
	113	8
	114	9
	
	210	10
	211	11
	212	12
	213	13
	214	14

C.Sc. 110: Computer Systems Index 2

Examples of Addressing Modes

ACC : 0

Load Immediate 110	110	210
	111	212
	112	7
Load Direct 110	113	8
	114	9
	
	210	10
	211	11
	212	12
	213	13
	214	14

C.Sc. 110: Computer Systems Index 2

Examples of Addressing Modes

ACC : 210

Load Immediate 110	110	210
	111	212
	112	7
	113	8
	114	9
	
	210	10
	211	11
	212	12
	213	13
	214	14

C.Sc. 110: Computer Systems Index 2

Examples of Addressing Modes

ACC : 0

Load Immediate 110	110	210
	111	212
	112	7
	113	8
Load Direct 110	114	9
	
Load Indexed 110	210	10
	211	11
	212	12
	213	13
	214	14

C.Sc. 110: Computer Systems Index 2

Examples of Addressing Modes **ACC: 0**

Load Immediate	110	210
	111	212
	112	7
Load Direct	110	113
		114
	
Load Indexed	110 + 2	210
		211
		212
		213
		214
		14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes **ACC: 7**

Load Immediate	110	210
	111	212
	112	7
Load Direct	110	113
		114
	
Load Indexed	110 + 2	210
		211
		212
		213
		214
		14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes **ACC: 0**

Load Immediate	110	210
	111	212
	112	7
Load Direct	110	113
		114
	
Load Indexed	110 + 2	210
		211
		212
		213
		214
		14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes **ACC: 0**

Load Immediate	110	210
	111	212
	112	7
Load Direct	110	113
		114
	
Load Indexed	110 + 2	210
		211
		212
		213
		214
		14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes **ACC: 10**

Load Immediate	110	210
	111	212
	112	7
Load Direct	110	113
		114
	
Load Indexed	110 + 2	210
		211
		212
		213
		214
		14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes **ACC: 0**

Load Immediate	111	210
	111	212
	112	7
	113	8
	114	9
	
	210	10
	211	11
	212	12
	213	13
	214	14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes **ACC : 111**

Load Immediate 111	110	210
	111	212
	112	7
	113	8
	114	9
	
	210	10
	211	11
	212	12
	213	13
	214	14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes **ACC : 0**

Load Immediate 111	110	210
	111	212
	112	7
Load Direct 111	113	8
	114	9
	
	210	10
	211	11
	212	12
	213	13
	214	14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes **ACC : 212**

Load Immediate 111	110	210
	111	212
	112	7
Load Direct 111	113	8
	114	9
	
	210	10
	211	11
	212	12
	213	13
	214	14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes **ACC : 0**

Load Immediate 111	110	210
	111	212
	112	7
Load Direct 111	113	8
	114	9
	
Load Indexed 111	210	10
	211	11
	212	12
	213	13
	214	14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes **ACC : 0**

Load Immediate 111	110	210
	111	212
	112	7
Load Direct 111	113	8
	114	9
	
Load Indexed 111 + 2	210	10
	211	11
	212	12
	213	13
	214	14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes **ACC : 8**

Load Immediate 111	110	210
	111	212
	112	7
Load Direct 111	113	8
	114	9
	
Load Indexed 111 + 2	210	10
	211	11
	212	12
	213	13
	214	14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes

ACC : 0

Load Immediate	111	110	210
		111	212
		112	7
Load Direct	111	113	8
		114	9
		
Load Indexed	111 + 2	210	10
		211	11
Load Indirect	111	212	12
		213	13
		214	14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes

ACC : 0

Load Immediate	111	110	210
		111	212
		112	7
Load Direct	111	113	8
		114	9
		
Load Indexed	111 + 2	210	10
		211	11
Load Indirect	111	212	12
		213	13
		214	14

C.Sc. 110: Computer Systems Index

Examples of Addressing Modes

ACC : 12

Load Immediate	111	110	210
		111	212
		112	7
Load Direct	111	113	8
		114	9
		
Load Indexed	111 + 2	210	10
		211	11
Load Indirect	111	212	12
		213	13
		214	14

C.Sc. 110: Computer Systems Index

Relocating Code: The Base Register

- If we want to run more than one program at a time we need to be able to relocate programs.
- If we can't then we'd have to write every single program such that they all used different addresses.
- Hardware support for relocation can be provided in the form of a base register.
- Value in the base register is added to each operand before address resolution.

C.Sc. 110: Computer Systems

The Base Register in Action

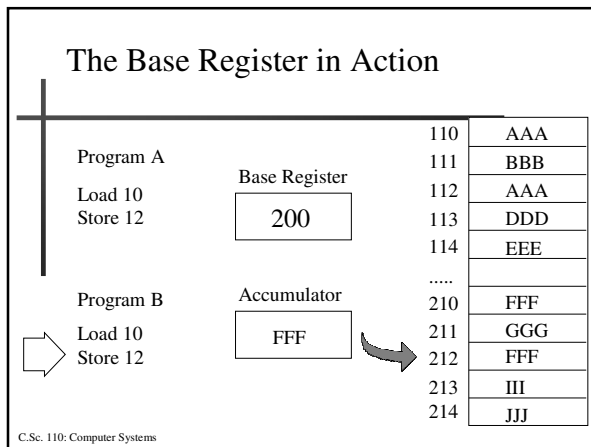
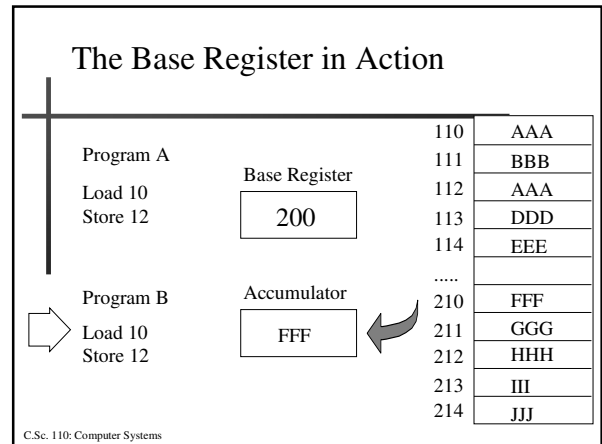
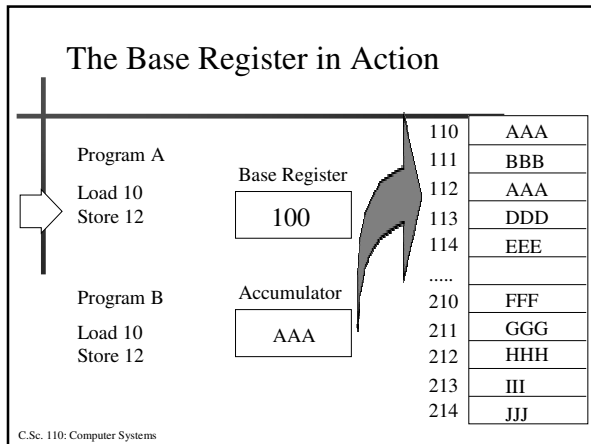
Program A		110	AAA
		111	BBB
Load 10		112	CCC
Store 12	Base Register	113	DDD
	<input type="text" value="100"/>	114	EEE
		
Program B		210	FFF
Load 10	Accumulator	211	GGG
Store 12	<input type="text"/>	212	HHH
		213	III
		214	JJJ

C.Sc. 110: Computer Systems

The Base Register in Action

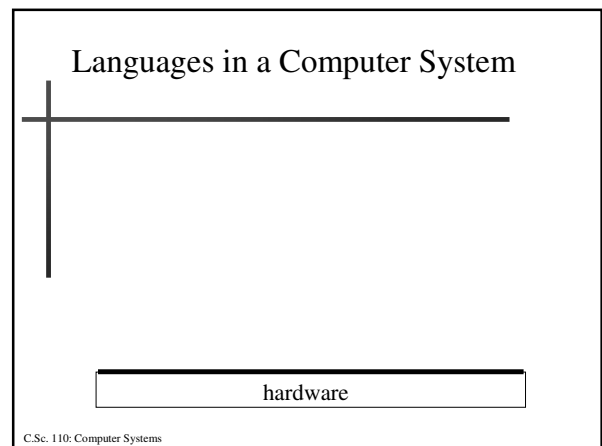
Program A		110	AAA
		111	BBB
Load 10		112	CCC
Store 12	Base Register	113	DDD
	<input type="text" value="100"/>	114	EEE
		
Program B		210	FFF
Load 10	Accumulator	211	GGG
Store 12	<input type="text" value="AAA"/>	212	HHH
		213	III
		214	JJJ

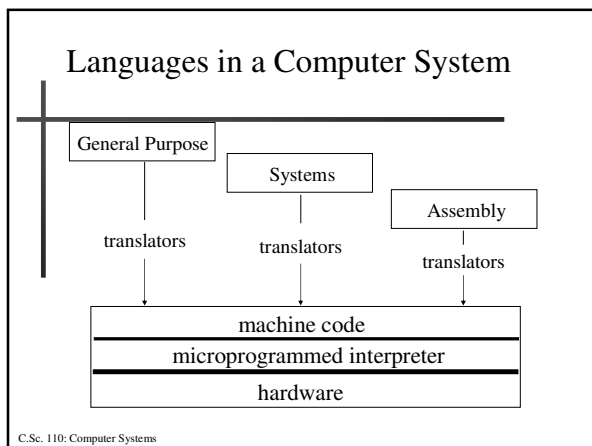
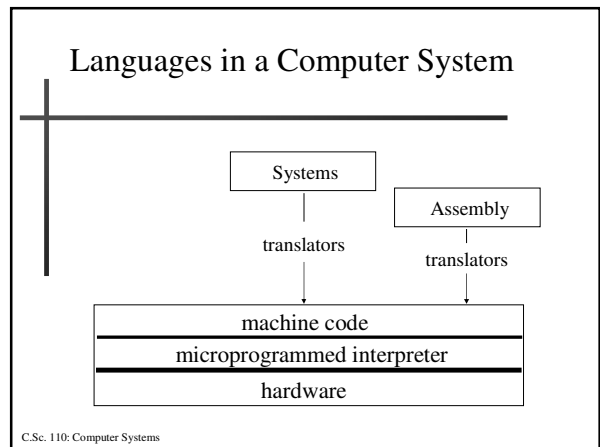
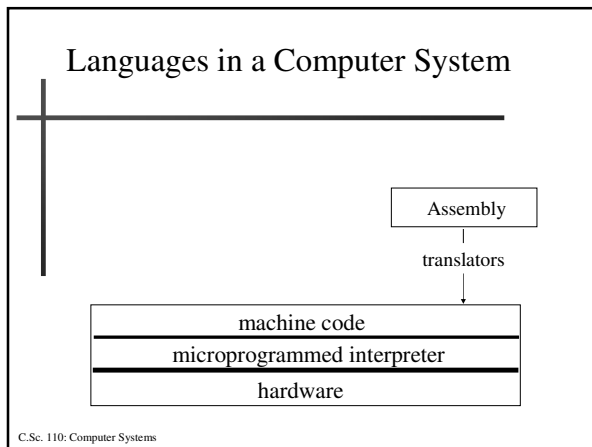
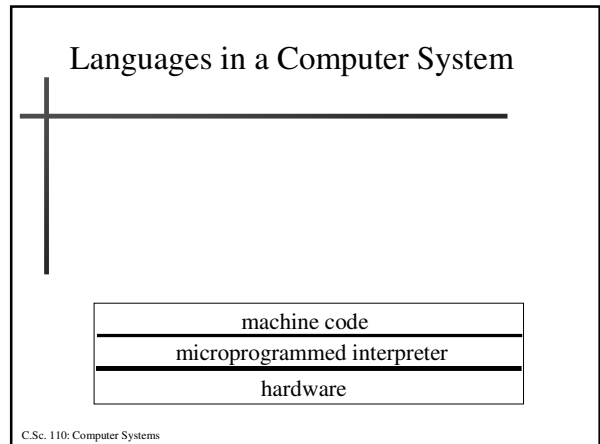
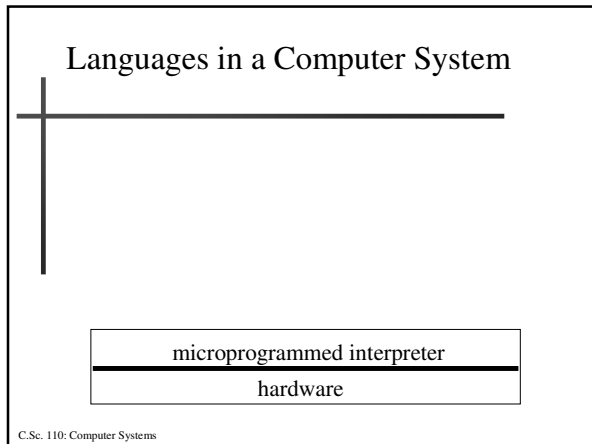
C.Sc. 110: Computer Systems



- ### Protecting Programs From Each Other
- If multiple programs are running we can check that they stay in their own sections of memory using a limit register.
 - Each store access is then checked against the limit register to ensure that it is within the programs allocated store.
- C.Sc. 110: Computer Systems

- ### Problems with Machine Code
- Large difference between the statement of a typical problem and its solution in machine code.
 - We have to convert all our operations and addresses into binary or hex patterns.
 - Every time we change the program we have to recalculate all the jump and data addresses.
 - Solution: use a higher-level language.
- C.Sc. 110: Computer Systems





- ### Assembly Languages
- Assembly languages are low-level languages which are very close to machine code.
 - In general one assembly language instruction = 1 machine code instruction.
 - Better support for operations: use mnemonics to make writing & understanding code easier.
 - Better support for data: provides support for writing code without having to hard-wire addresses into the code.
- C.Sc. 110: Computer Systems

An Assembly Language

- Each statement has four fields:-
 - a label field, a symbol to represent the address of the instruction or data.
 - an operation field which holds the name of an instruction.
 - an operand field with optional symbols to denote addressing modes.
 - a comment field.

C.Sc. 110: Computer Systems

An Assembly Language ... contd.

```
label: operation  mode operand  comment
```

C.Sc. 110: Computer Systems

An Assembly Language ... contd.

```
label: operation  mode operand  comment
here: LDA        #    50          ; load 50 into acc
there: STA              50          ; store A into loc. 50
      JMP              there
```

C.Sc. 110: Computer Systems

An Array Processing Program

```
      LDA    #50
      STA    COUNT      ; store iteration count
      LDA    #0
      STI    #0          ; set index register to 0
      STA    TEMP        ; initialise sum of elements
TOP:   LDA    TEMP
      ADD    ARRAY+
      STA    TEMP        ; add the next value
      LDI    #1
      ADD    #1
      STI    #1
      LDA    COUNT
      SUB    #1
      STA    COUNT
      JNZ   TOP
```

C.Sc. 110: Computer Systems

Assessment of Languages Covered

- Microcode
 - easy to build computers to execute microcode.
 - difficult to program in because instructions do too little.
- Machine code
 - easier to use than microcode.
 - difficult to write flexible code, programmers must work in hex & instructions still do very little.
- Assembly Language
 - more powerful and flexible than machine code or microcode.
 - lack of portability and still a long way from the problem.

C.Sc. 110: Computer Systems

Summary

- Described extensions to our machine code's addressing scheme to make it more powerful.
- Covered the relationship between high-level languages and machine languages/code in the system.
- Described a simple assembly language.
 - Still relevant today – check out job adverts for PS2 programmers

C.Sc. 110: Computer Systems

Coming Next Week

- Language translation and Java...
- D&L pages 235-241