

Lecture 21

Intro to Assembly language

- Add, R2, R0 is an assembly language instruction for our simple processor
- Many different assembly languages
- Human readable (one step away from binary)
- Compiler: translates high-level language into assembly
 - User does not have to worry about the version of assembly the computer uses
- Each instruction specifies
 - Operation to perform (e.g. Add, Sub)
 - Operands to operate on (values in register, memory, constants)

Intro to Computer Organization

- Processor
 - Stores address to data in memory
 - Data in/out
 - Read/write All of these are connected to the 32 bit bus. Bits are allocated to the address, data in/out (separately), read/write (separately), and there are I/O ports.

Summary memory and I/O ports are each assigned a range of addresses called memory map - e.g. memory: 0x0 - 0x3FFFFFFF, LEDR 0xFF200000 - 0xFF20000F - only one device responds to unique address from processor

Memory Architecture

- Similar to 2D array you index into
- e.g. word = 32 bits = 4 bytes (byte addressable memory)
 - address 0 = word 0
 - address 8 = word 2
 - 256 bit address can store 64 words (2 bits are used to select the byte within the word)

Wrap up

- Processors are built out of digital logic building blocks
- Processors execute assembly language instructions
- Processors are organised with memory and other I/O peripherals

Lab 6

- Most non trivial digital circuits are separated into 2 main functions
 - Datapath: where data flows (registers, muxes, ALU, etc)
 - Control path: manipulates signals in datapath to control operations and how data flows e.g. mux select signals, register enables