EECE 218
Microcontrollers

The von Neumann Computer
von Neumann Computer

Memory:

- Array of storage cells / $n$ bits per cell (HC12: $n=8$)
- Implemented in solid-state
  - (D) f/f = Static R(andom)A(ccess)M(emory)
  - Capacitor = D(ynamic)RAM
  - ‘Fuse’ = P rogammable R(ead) O(nly) M(emory)
  - Floating gate = E(rasable)PROM
Memory

CPU asserts ADDRESS and R/~W signal:
after a brief delay a binary word on the DATA lines is either READ FROM or WRITTEN into the selected CELL.
Memory devices

- ROM: Read-only/non-volatile,
- RAM: read-write/volatile
- Flash: ROM, but programmable
- EEPROM: ROM, but programmable

All of the above: on the 9S12 chip!

For our 9S12:

#(Address) = 16 bits, #(data) = 8 bits
Not all 64kbytes is usable on 9S12 (we will use 12 kbytes RAM segment in the lab)

Memory is used to store:

- Instructions: binary codes prescribing instructions
- Data: to be manipulated

Instruction and Data areas in memory must be separated.

In machine code programming:

It is the programmer’s responsibility.
von Neumann Model - Details

CPU
- Program Counter
- Arithmetic/Logic Unit
- Registers
- Controller

Memory
- Address
- Data
- Control
von Neumann Model - Details

- **Program Counter:**
  » Contains the memory address of (‘points to’) the **next** instruction to be executed

- **Registers:**
  » Small, fast memory cells for storing intermediate results
  » ‘Accumulator’ for data, ‘Index register’ for address

- **ALU: Arithmetic/Logic Unit**
  » Circuit to implement binary arithmetic/logic/shift....
von Neumann Model – Instruction cycle

1. FETCH: Send PC to address bus & fetch instruction code, update PC
2. DECODE: Interpret instruction code
3. [FETCH DATA: fetch extra operands, update PC]
4. EXECUTE instruction
5. REPEAT

Notes:
- Memory cycle = 1 transfer from/to memory
- Instruction cycle = 1 or more memory cycles
- Instruction contains: instruction + [optional: operands]
Machine code programming

In high-level languages: data is in variables.
In machine code:
  data is in memory location(s) or in register(s)

Must learn to distinguish:
  address (or location) of data --- where it is
  value (or content) of data --- what it is