Review and AVR Architecture

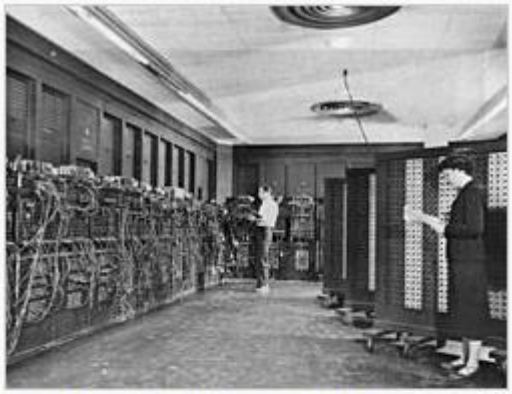
Microcontrollers and AVR Specific Information

Instructor: Tinoosh Mohsenin

Credit to Dr. Robucci for original slides

Background on general purpose computer

- ENIAC: Electronic Numerical Integrator And Computer was the first general-purpose computer
  - Built in Univ. of Pennsylvania
  - Funded partially by US Army 1943 before ending WWII
- Von Neumann computer
  - 1944



Glen Beck (background) and Betty Snyder (foreground) program ENIAC in BRL building 328. (U.S. Army photo)

Programmers Betty Jean Jennings (left) and Fran Bilas (right) operate ENIAC's main control panel at the Moore School of Electrical Engineering. (U.S. Army photo from the archives of the ARL Technical Library)

Source: wiki

{micro} Processor, Computer, Controller

- CPU
  - Unit that fetches and processes a set of general-purpose instructions
- Microprocessor
  - A CPU on a single chip. It may also have other units (e.g. caches, floating point processing)
- Microcomputer
  - A microprocessor + I/O + memory+ etc are put together to form a small computer for applications like data collection, or control application.
- Microcontroller
  - A microcomputer on a single chip. It brings together the microprocessor core and a rich collection of peripherals and I/O capability.

Microcontroller (MCU)

- Common peripherals include
  - Serial communication devices
  - Timers, counters, pulse width modulators
  - Analog-to-digital and digital-to-analog convertors
- Particularly suited for use in embedded systems
  - Real-time control applications
  - On-chip program memory and devices
- Enables single-chip system implementation
  - Smaller and lower-cost products
- Examples: Motorola 68HC11xx, HC12xx, HC16xx, Intel 8051,80251, PIC 16F84, PIC18, ARM9, ARM7, **Atmel AVR, etc.**

Slide info courtesy of Dr. Patel, CMPE310

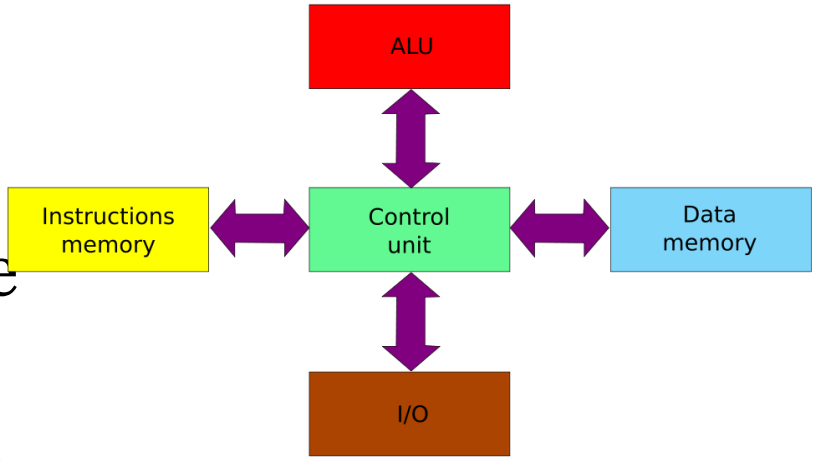
Alternatives for MCU

- Discrete ICs
  - Dedicated digital circuit
  - Can use various ICs for functions (AND, OR, etc..)
- PLD(Programmable Logic Device)
  - Contains various user selected logic functions
  - Results in more compact system compared to dedicated digital circuits
- ASIC(Application Specific Integrated Circuit)
  - Specific optimized implementation

Why use Microcontrollers?

- Peripheral loaded
  - ADC, DAC, GPIOs, Serial Interfaces
- Cheap
  - ~$1 for 8-bit processor
- Low Power
  - ~300μA operation (1 AA battery for 275 days)
  - <1μA sleep (1 AA battery for 225 years)
- Programmable
  - Assembly or C

AVR Architecture



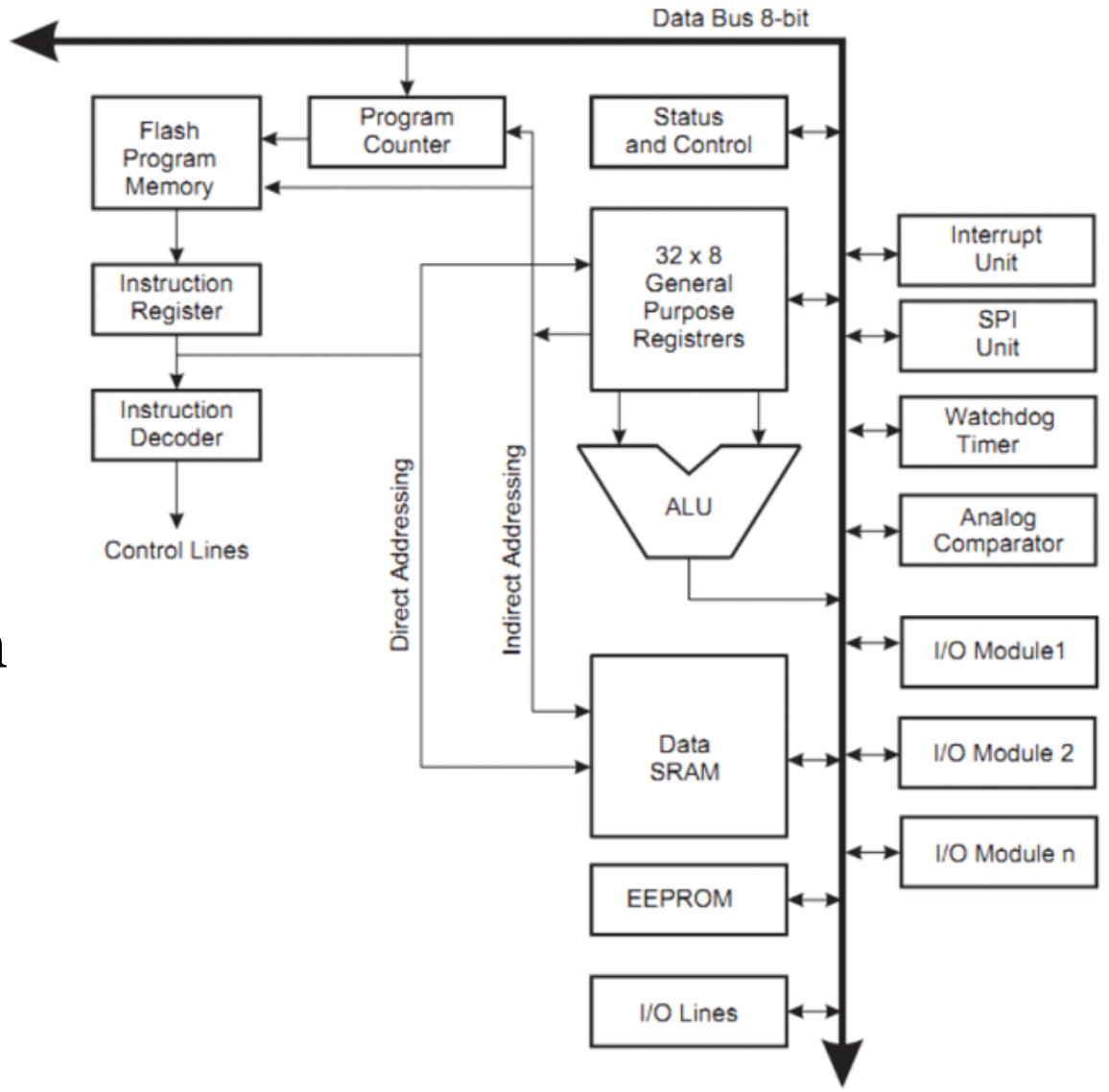
- RISC Harvard Architecture
  - RISC vs. CISC
  - Harvard vs. Von Neumann
    - Separate program and data memory bus
    - On-chip program memory → Flash memory
    - On-chip data memory → RAM and EEPROM
- 32 x 8 general purpose registers
- Internal and external interrupt sources
- On-chip RC clock oscillator
- Variety of I/O, Programmable I/O Lines

General 8-Bit AVR Architecture

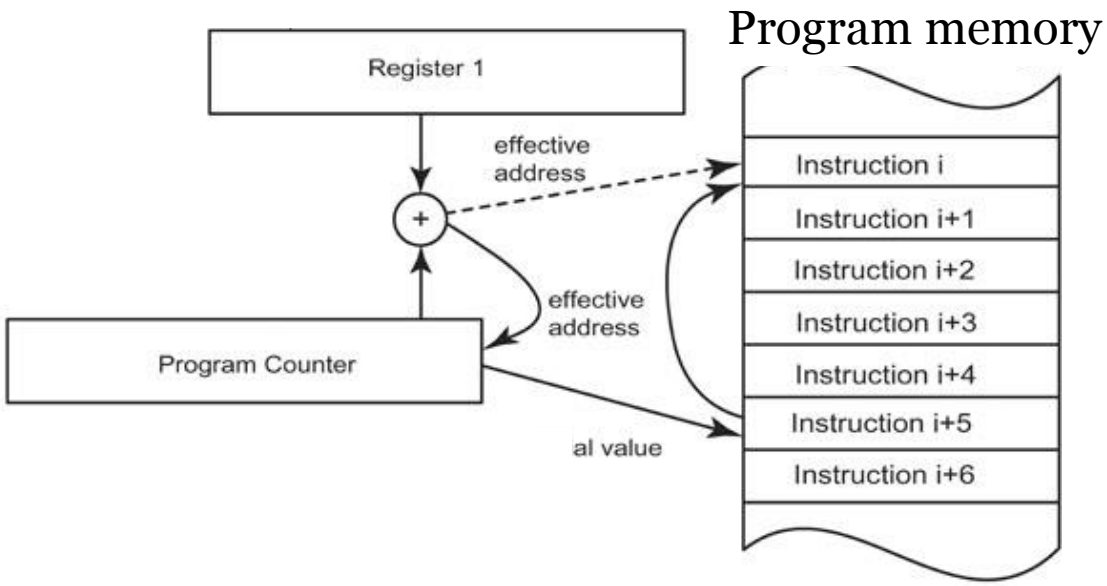
- Important block diagram to review
- ALU supports arithmetic and logic operations between registers or between a constant and a register.

- ALU connection with registers makes ALU execution with registers in one cycle

Figure 6-1. Block Diagram of the AVR Architecture



Program counter and instruction



Pipeline

- AVR has 2-stage pipeline (Fetch, Execute)
  - Decode occurs in IF cycle
- While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle.
- Most instructions execute in a single clock cycle
  - Some might take 2 or more cycles

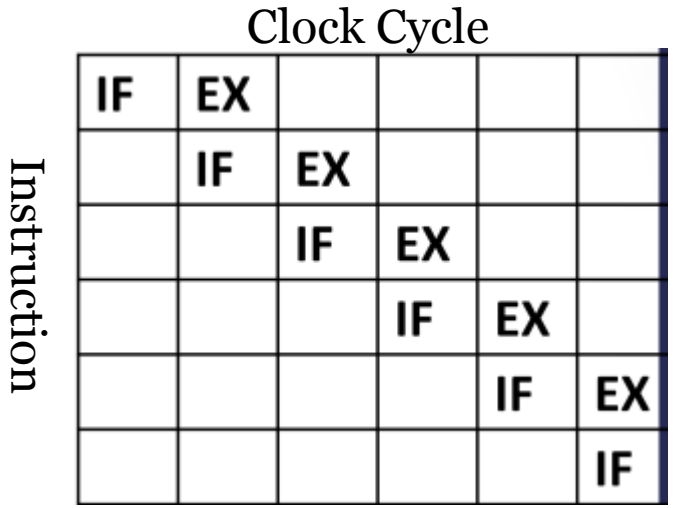
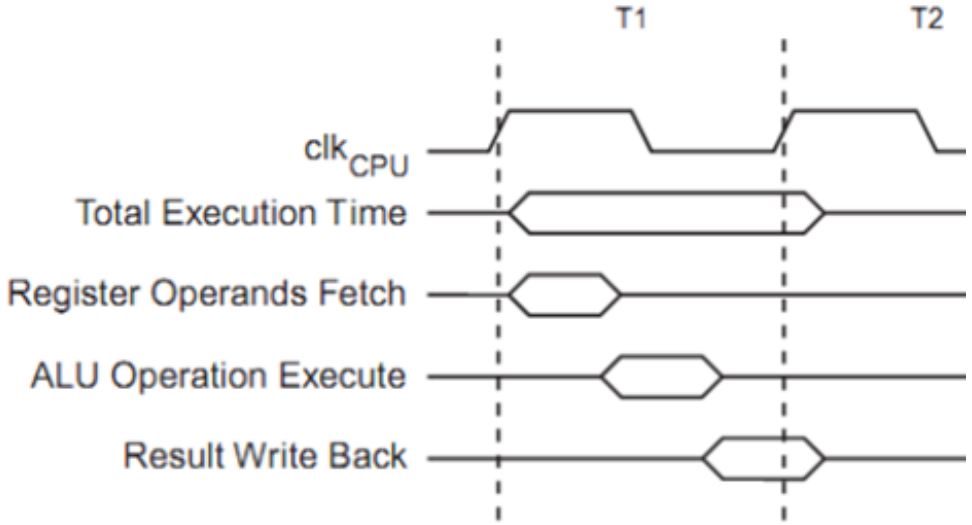


Figure 6-3. Single Cycle ALU Operation

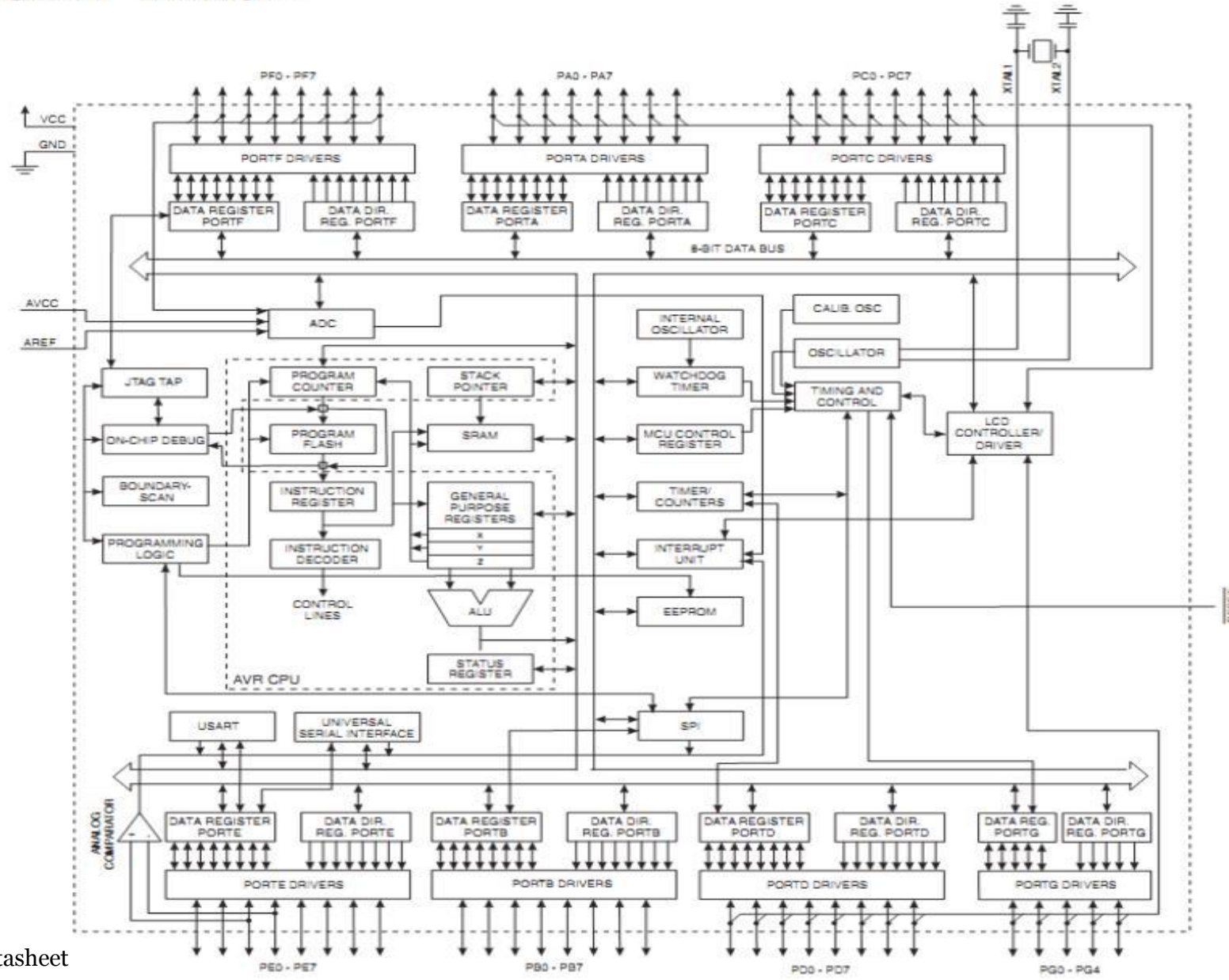


ATmega 169P

- Datasheet can be found at: http://www.atmel.com/Images/doc8018.pdf
  - From megaAVR family
    - Families – Similar group of devices
  - "P" of 169P means "Low Power"
  - "16" in 169P means 16k program memory
  - "9" in 169P means 9th design revision

Atmega 169P

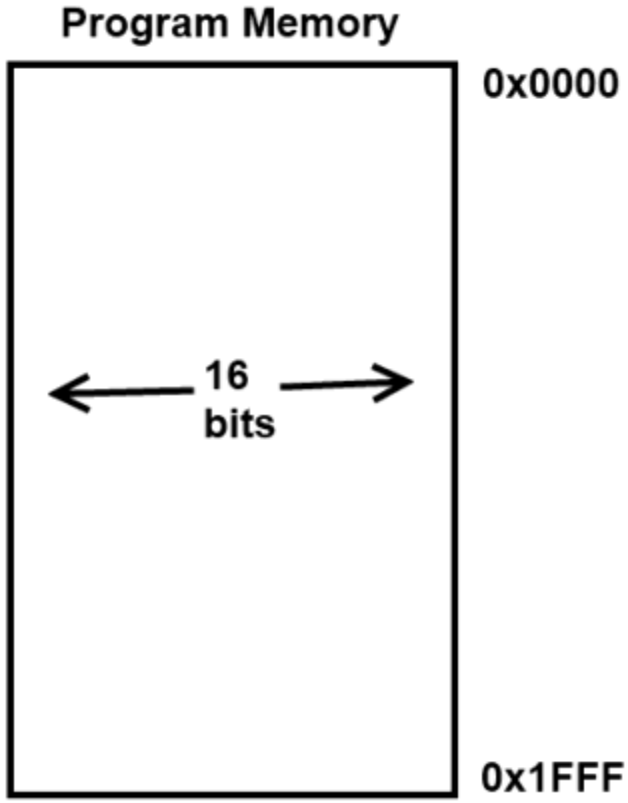
Figure 2-1. Block Diagram



Source: Atmega 169P datasheet

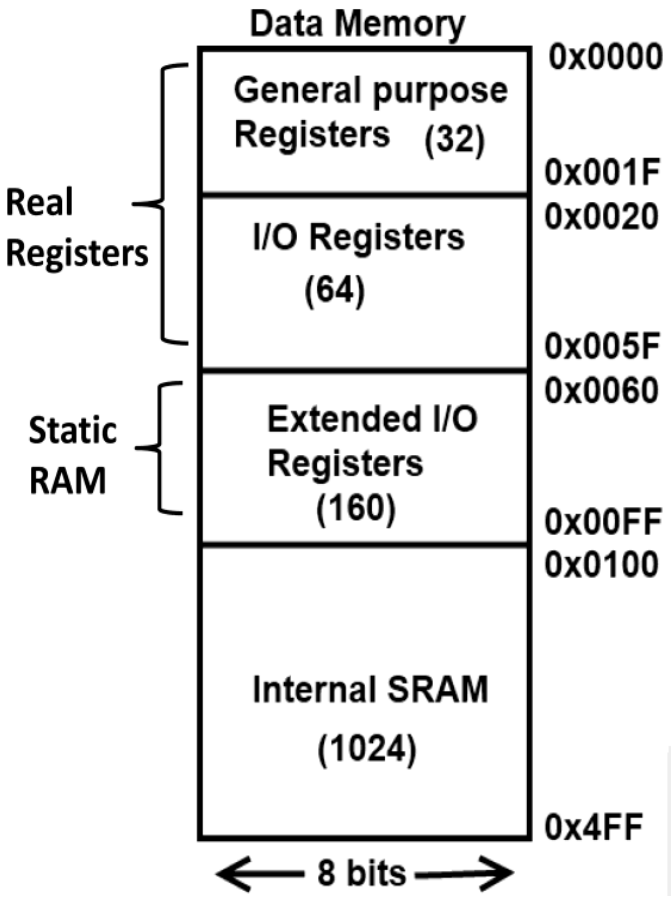
Program Memory

- Atmega 169P contains 16k bytes Flash memory
  - Program storage
    - Boot program section and Application Program Section
- All AVR instructions are 16 or 32 bits wide
- Flash organized as 8k*16
- 8k = 2^13 → 13 bits needed to address program memory
  - Program Counter = 13 bits



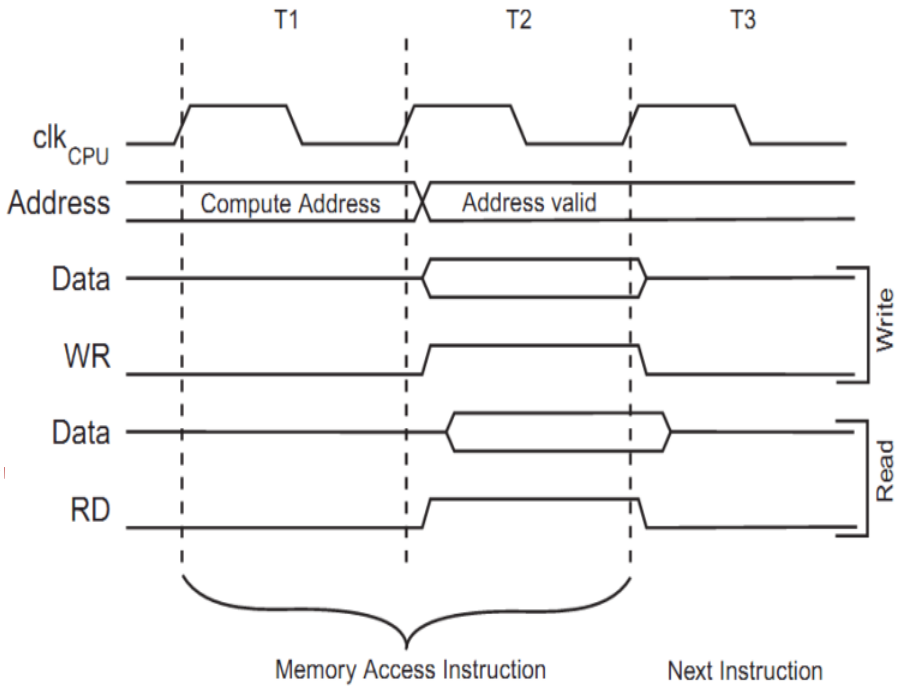
Data Memory

- Entire data memory space can be accesse as memory
  - Register instructions faster on real register
- General Purpose Registers
  - 32, labeled R0-R31
  - Called Register File
  - Can be used in instructions
    - i.e. ADD R2, R3
- The lower 1,280 data memory locations address both the Register File, the I/O memory, Extended I/O memory, and the internal data SRAM. The first 32 location address the Register File, the next 64 location the standard I/O memory, then 160 locations of Extended I/O memory, and the next 1024 locations address the internal data SRAM.
- Refer to section 7.2.1



SRAM Access Time

- 2 Cycles to access SRAM
- Address pointers (Registers X,Y,Z) used for addressing
- First Cycle
  - Register file accessed
  - ALU calculates address
- Second Cycle
  - Calculated address accesse SRAM location
- Refer to section 7.2.1



EEPROM Space

- EEPROM
  - Electrically erasable programmable read-only memory
  - 100k write/erase cycles for a lifetime
  - 512 bytes
- Persists after power-down
- Reads and Writes will halt CPU
- For more information on EEPROM Refer to the slides Memory Types

AVR Reset resources

The ATmega169P has five sources of reset:

- 1. Power-on Reset. The MCU is reset when the supply voltage is below the Power-on Reset threshold (VPOT).
- 2. External Reset. The MCU is reset when a low level is present on the RESET pin for longer than the minimum pulse length.
- 3. Watchdog Reset. The MCU is reset when the Watchdog Timer period expires and the Watchdog is enabled.
- 4. Brown-out Reset. The MCU is reset when the supply voltage VCC is below the Brown-out Reset threshold (VBOT) and the Brown-out Detector is enabled.
- 5. JTAG AVR Reset. The MCU is reset as long as there is a logic one in the Reset Register, one of the scan chains of the JTAG system. Refer to the section ”IEEE 1149.1 (JTAG) Boundaryscan” on page 259 for details.