EE2016 Microprocessor Theory & Lab
Aug – Nov 2024

Week6: Control Unit Anatomy: Hardwired Control Unit

Dr. R. Manivasakan, OWSM Lab, EE Dept, IIT, Madras

Manual Execution of a Program

- Manual execution of program (not automated)
  - We manually identify the next instruction to be executed, drive the EU using the right control signals
- Automation is done by CU.
  - Uses the PC (part of CU) to hold address of next instruction to be executed.
  - And also the PFCL would also automatically track the next instruction to be executed even of the sequence is out of order
  - No human intervention is needed
  - Automatic

Control Unit Design

- We have designed an EU so far and learned how to control EU using control lines: we, the human being interpreting the program, generate control signals and drive the EU
  - Now, let us learn how to automate the program to drive the EU

We have so far learned .....

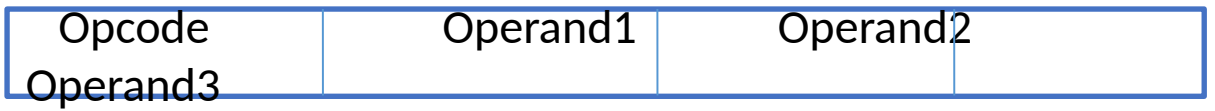
- How to design EU of a processor?
- How to drive EU using Control Signals?
- Running an assembly program
  - involves generating the appropriate control signals given a program (and instructions thereof)
  - The generated control signals now drive the EU
  - Normally the program is executed in sequence
  - Occasionally change sequence (program flow) as per the outcome of previous instruction(s)
- Next we will learn
  - How to automate the controller (so far a human being a controller) using a Control Unit and Program Memory

Control Unit

- **Control unit** (CU) uses a Program to control/drive the operation of the Execution unit (EU) of the processor apart from controlling peripherals including memory
  - Each program instruction (Execution Unit Control Instruction or EUCI) is executed as a series of steps in sequence driven by a state machine
    - Drives data-flow and ALU instructions by driving Control signal of EU
    - We can say that the control unit takes charge of running the given program, all by itself: Automates the execution of a program
    - Implemented using a binary decoder to convert instructions into timing and control signals that direct the operation of the EU, I/O and memory.
      - Dictates what has to be done in each clock period.
  - Program is normally executed in Sequence, but some instructions can modify the flow of the program (Program Flow Control Instruction) – rather the outcome of some instructions (conditional statements) might alter the course of the program
- CU Components: Instruction Decoder, Program Counter (PC), Stack Pointer (SP), Execution Unit Control Logic (EUCL) and Program Flow Control Logic (PFCL)

Instructions and Programs

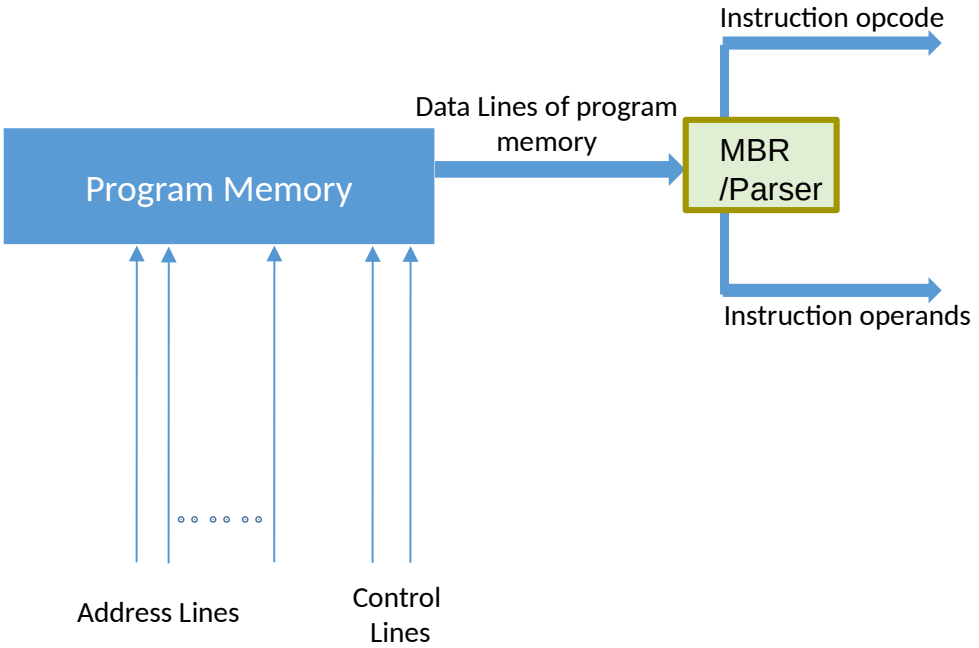
- Programs are sequence of instructions
  - to achieve a task
  - Each instruction requires the microprocessor to execute a primitive function
  - Communicate with the processor through instructions
- Each instruction has



  - Opcode: Operation
  - Operands: The data that would be required to execute the instruction. The number of operands required depends on the opcode of the instruction
- Program: An organized list of instructions that, when executed, causes the processor to behave in a predetermined manner to accomplish a task
- Instructions in a program can be broadly categorized as
  - Execution unit control instructions: controls the execution unit
  - Program flow control instructions: Alter flow of the program depending on Flag bits
- Processor Instruction Set: defined by the manufacturer

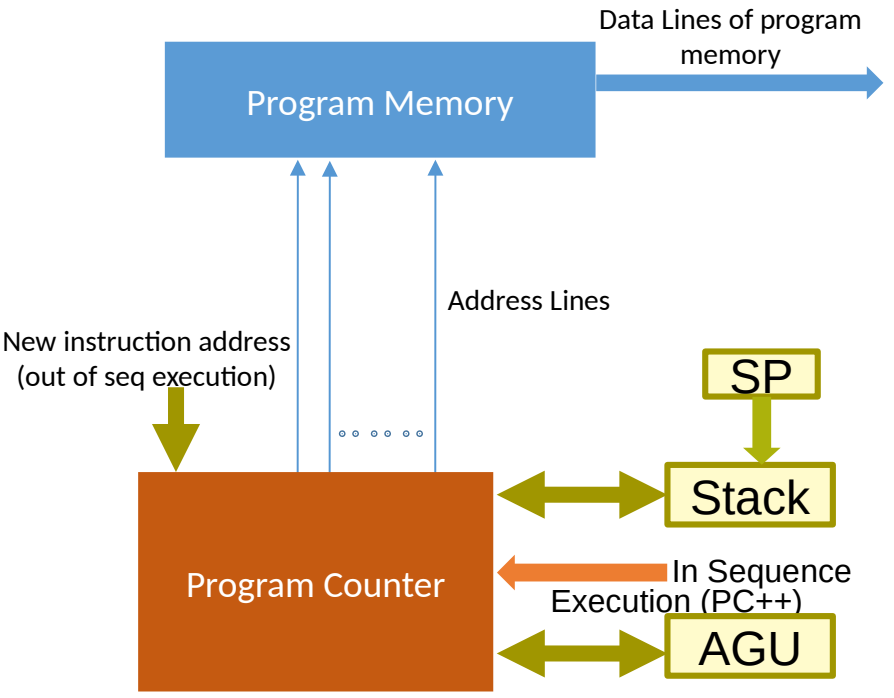
Program Memory (PM)

- PM is used to store a program
  - Each instruction of the program in PM is stored at consecutively addressable locations
- Address lines and control lines of PM helps fetch next instruction to be executed from the memory
- Data Lines of PM gives opcode and the operand (In AVR program data bus is 16 bits wide) and is unidirectional.



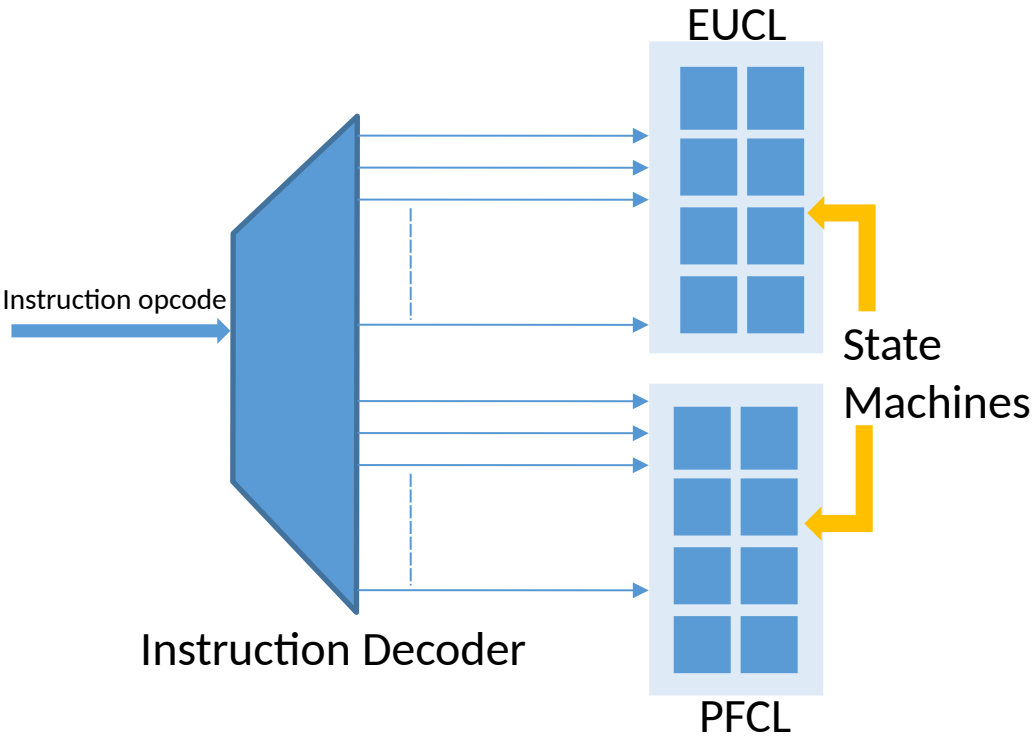
Program Counter (PC)

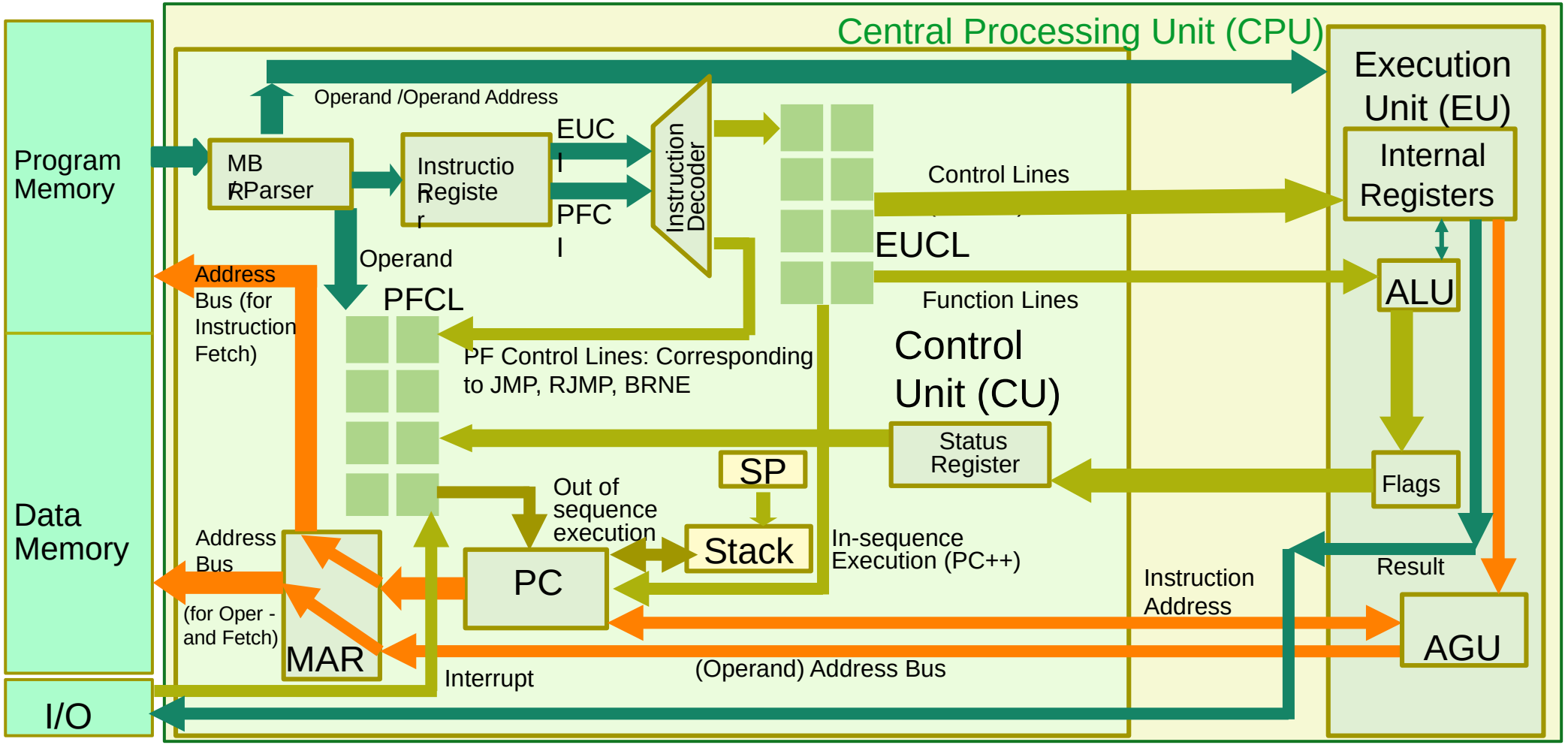
- PC holds the address of the next instruction to be executed and drives the address lines of PM
- Instructions are usually fetched sequentially from memory (hence PC is usually incremented after each instruction is executed)
  - but **program flow control instructions** might change the sequence by loading a new value in PC



EUCL, PUCL and Instruction Decoder

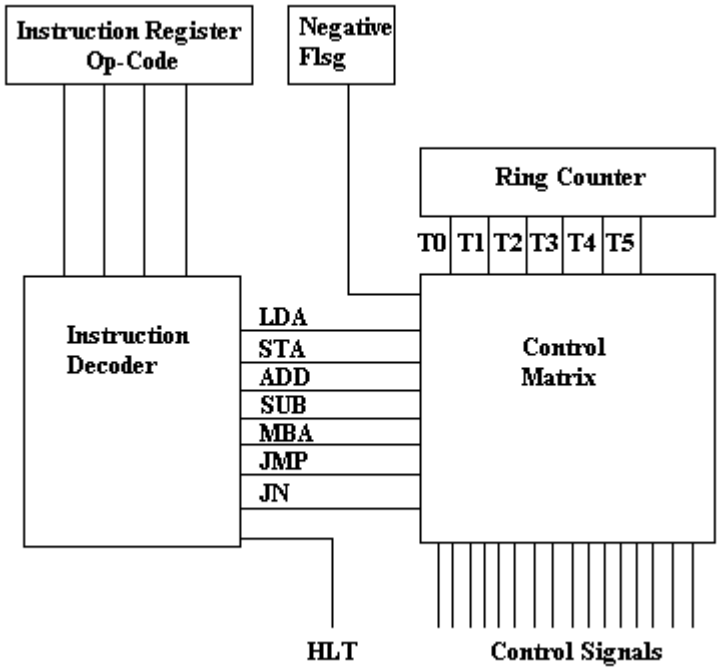
- An Instruction Decoder decodes the op-code and drives the state machine corresponding to the op-code
  - Execution Unit Control Logic (EUCL): issues control signals for the execution unit




Central Processing Unit (CPU)
Program Memory
Data Memory
I/O
Execution Unit (EU)
Internal Registers
ALU
Flags
AGU
Operand /Operand Address
MB RParser
Instructio Registe r
EUC I
PFC I
Instruction Decoder
Control Lines
EUCL
Function Lines
Operand
PFCL
Address Bus (for Instruction Fetch)
PF Control Lines: Corresponding to JMP, RJMP, BRNE
Control Unit (CU)
Status Register
SP
Out of sequence execution
Stack
In-sequence Execution (PC++)
Address Bus
(for Oper - and Fetch)
MAR
PC
Interrupt
(Operand) Address Bus
Instruction Address
Result

Hard Wired Control Unit

- Two Approaches for implementation of CU
  - Hardwired control unit
  - Microprogram control unit
- Hardwired control unit
  - Implemented through use of combinational logic units, which can generate the electrical control signals corresponding to an input
  - Generally faster than microprogrammed approach
  - Inherently not flexible. Fixed at the design stage itself.
- Microprogram control unit
  - Microprogram consists of microinstructions and stored in special control memory
  - A single instruction in assembly might consist of a microprogram (which consists of many microinstructions).
  - Much more flexible than hardware control unit.

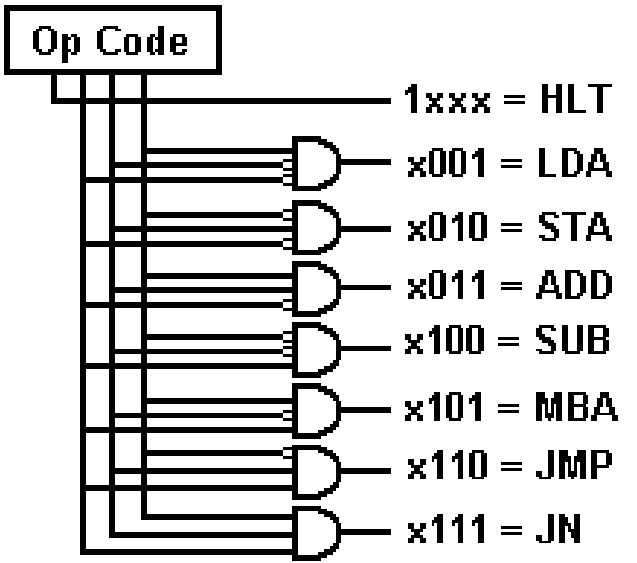


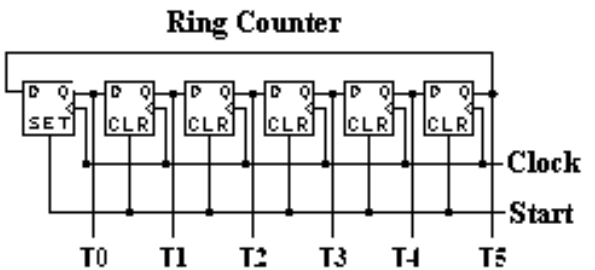
Constant length OP-Codes are assumed

https://illustratedman-code.github.io/HardVsMicro/

Instruction Decoder

- Instruction Decoder
  - Receives the opcode sent to it by the IR
  - Interprets the opcode as a specific signal
  - Sends a signal to the control matrix corresponding to the opcode from the IR Each line to the Control matrix represents a different signal (a set of pins)
- Rather than ring counter, a ring pulse generator, generated disjoint consecutive pulses,
  - Whose edge triggers the operations of raw control signal (low level instructions)
  - Examples of low level instructions (raw control signals) are IP, LP, EP etc



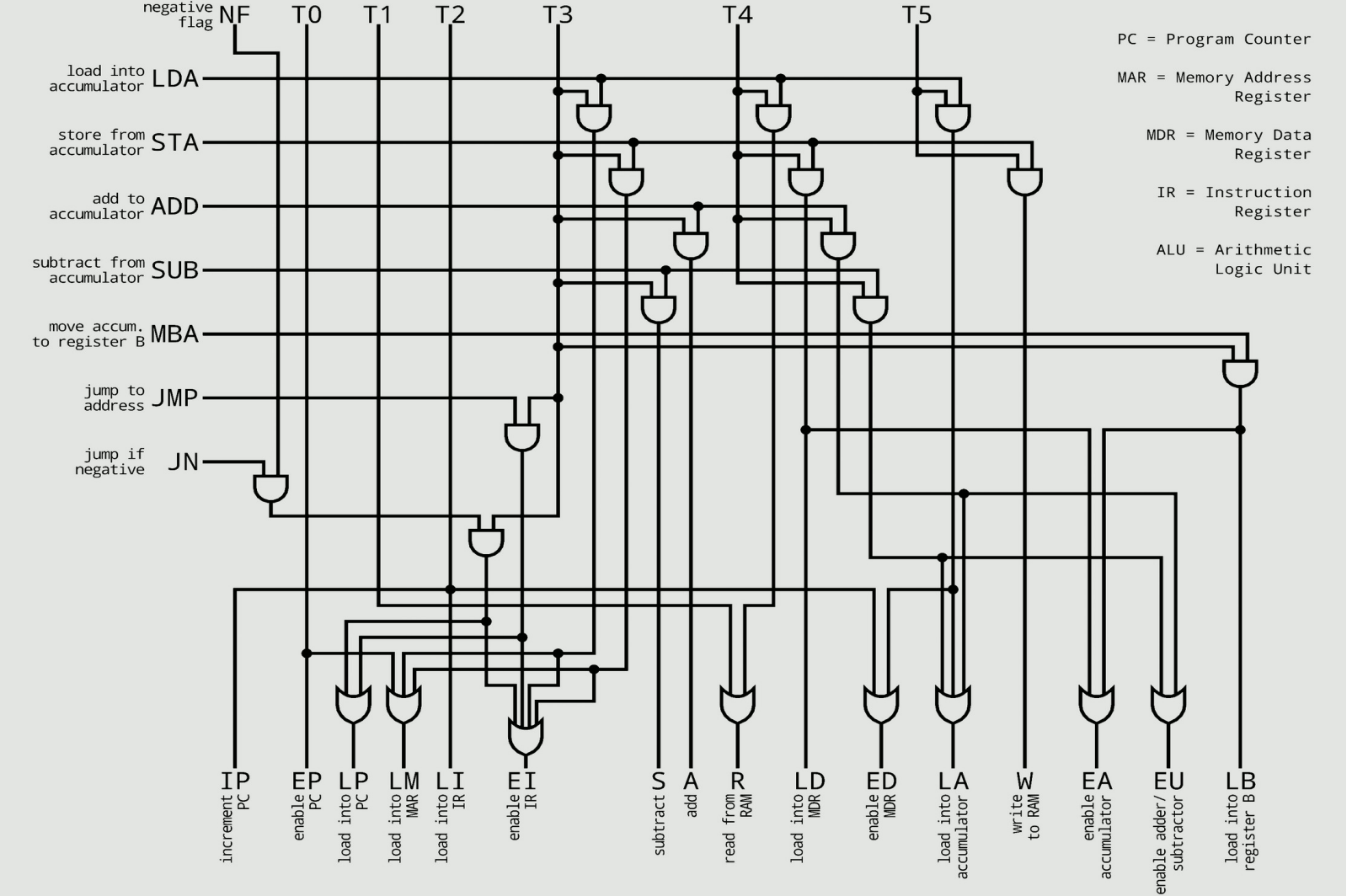


Hard Wired Control Unit - Matrix

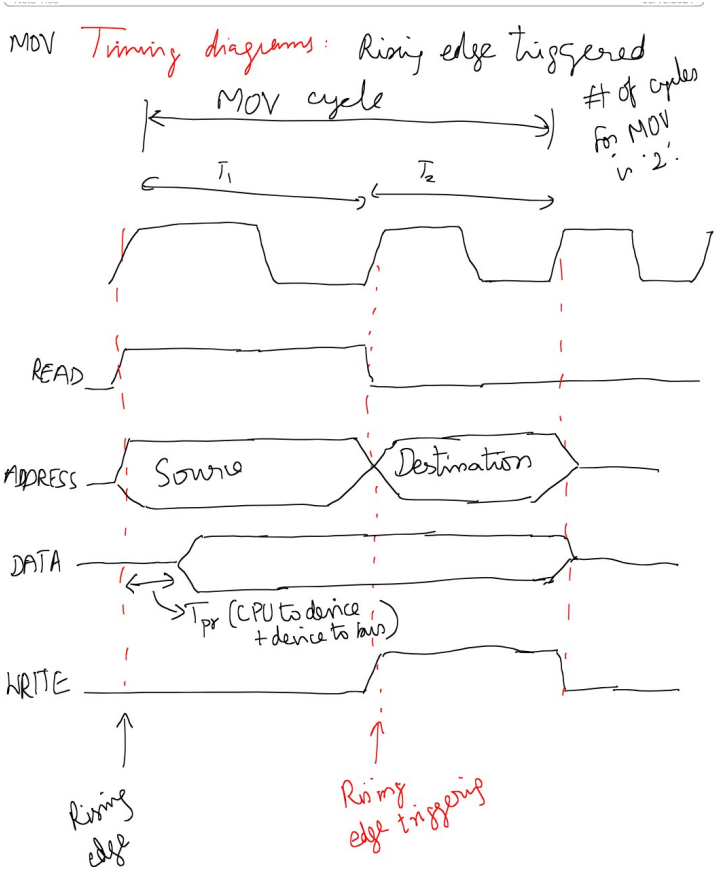
- Low level instructions (raw control signals):
  - R, W, EP, IP, LP, EI, LI, S, A, EA, LA etc
- Examples of dissecting an AVR assembly instruction into low level instructions
- `MOV Rd, Rs`
  - Step 1: Read from source register ‘Rs’, its contents into the data bus, R – read from RAM
  - Step 2: Write the contents of data bus into the destination register ‘Rd’ – W - write into RAM
- `ADD Rd, Rs`
  - T4 (EU, LA), T3 (A)
-

Hardwired Control Unit

Matrix part



Example: MOV


MOV Timing diagrams: Rising edge triggered
MOV cycle
# of cycles for MOV is 2.
T1
T2
READ
ADDRESS
Source
Destination
DATA
Tpr (CPU to device + device to bus)
WRITE
Rising edge
Rising edge triggering