Course Overview

`http://www.ee.nmt.edu/~rison/ee308`

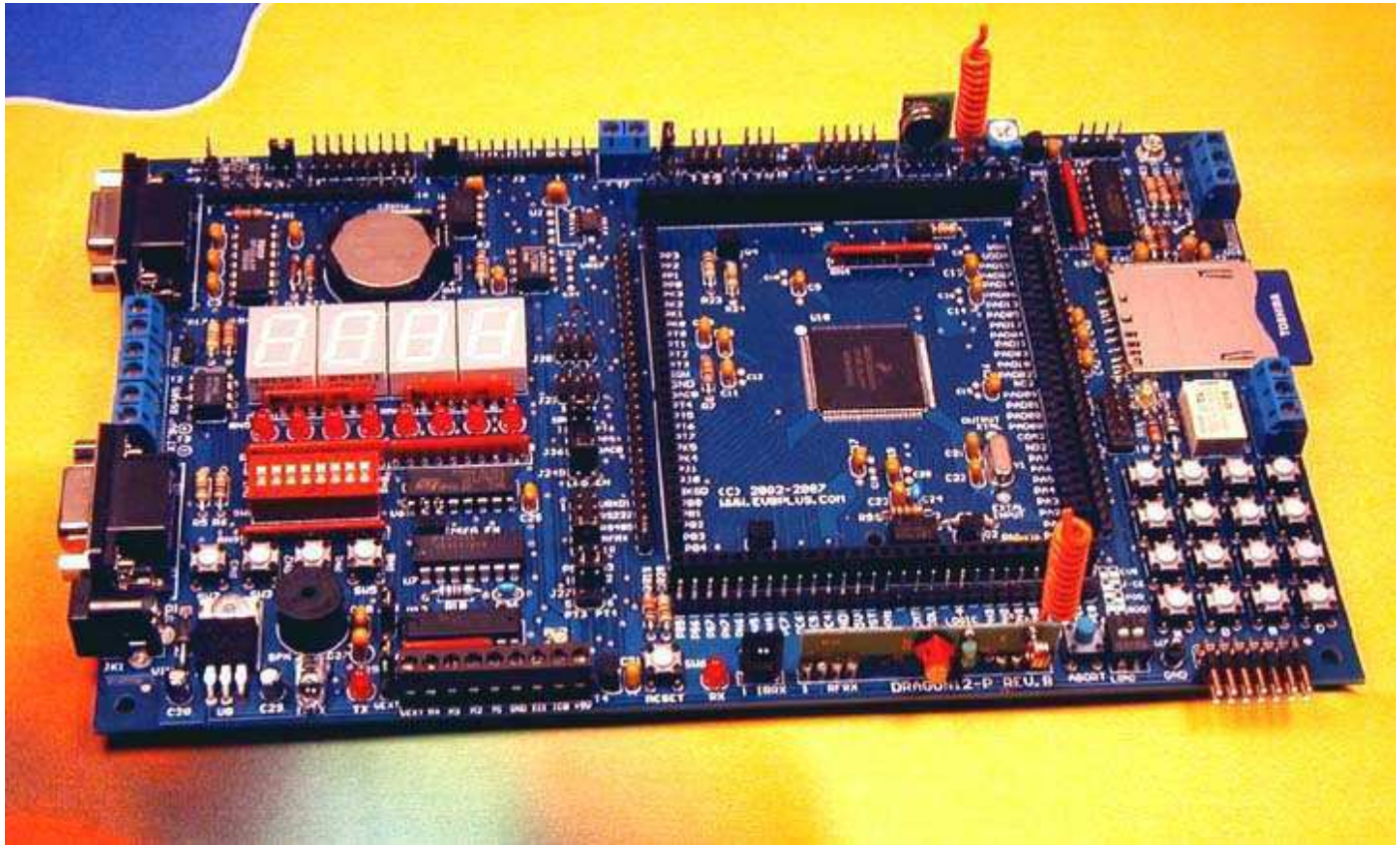
Texts: **Freescale Databooks on the MC9S12**
You can pick up the Freescale data sheets at the end of class today
The HCS12/9S12: An Introduction to Software and Interfacing

Grading:

- 20%: Homework due every Monday.
- 10%: Quiz every Friday
- 45%: Four midterms exams
- 25%: Final exam

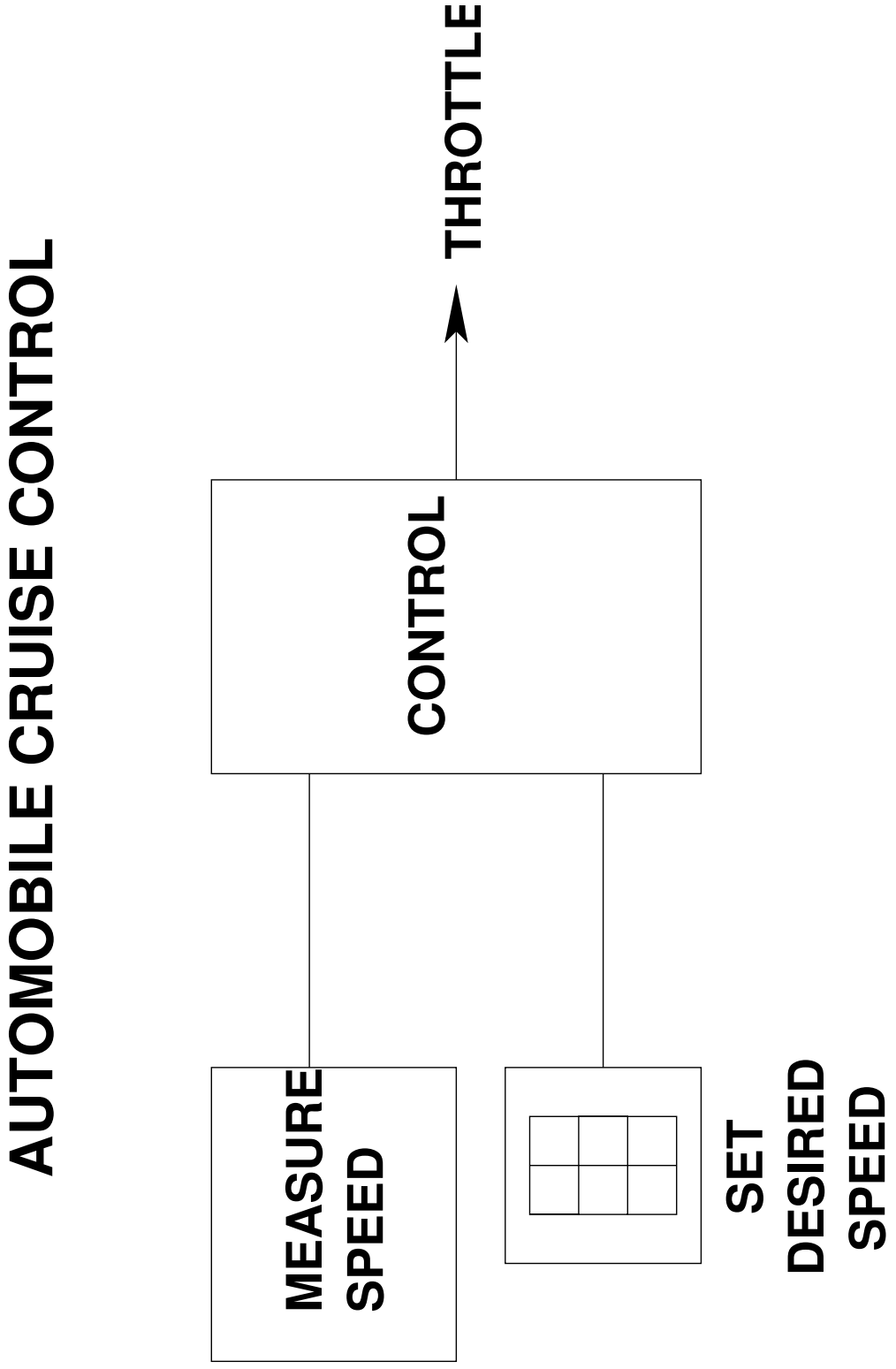
Outline:

- Introduction to the MC9S12 Microcontroller
- Binary and Hexadecimal Numbers
- Assembly Language Programming
- C Language Programming
- Introduction to MC9S12 Internal Peripherals
  - The MC9S12 Timer Subsystem
  - Interrupts using the Timer Subsystem
  - The MC9S12 Pulse Width Modulator Subsystem
- The MC9S12 Expanded Mode
  - Address and Data Buses and Timing
  - Adding Memory and External Peripherals
  - Interfacing to the MC9S12
- More MC9S12 Internal Peripherals
  - The A/D Converter Subsystem
  - The Serial Peripheral Interface
  - The Serial Communications Interface
- Using the MC9S12 in a Control Application

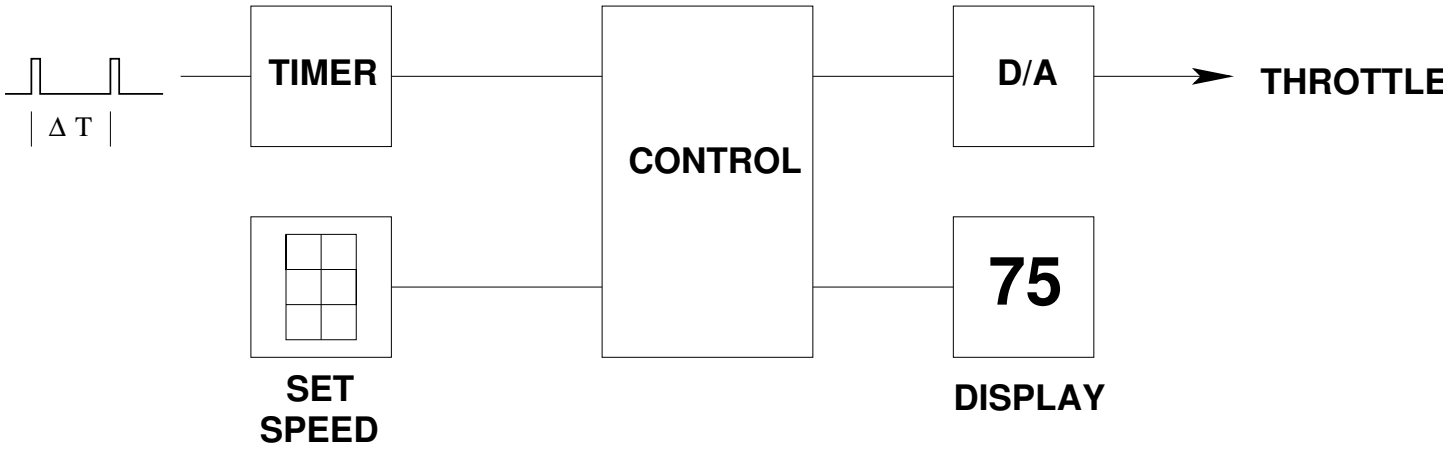


Lab Overview

- The lab instructors ar Dr. Aly El-Osery and Dr. Hector Erives. The lab meets Monday and Wednesday afternoons. **No labs this week.**
- Lab handouts will be posted on the EE 308 website: http://www.ee.nmt.edu/~rison/ee308
- The MC9S12 evaluation kits will be passed out in lab next week.
- **You need to bring a bound lab notebook to the first lab.**
- There will be a prelab for each lab. This must be done and turned in at the start of your lab section. The lab TA will verify that you have completed the prelab.
- Be prepared to answer questions about the pre-lab when you come to lab.
- If you do not complete the prelab before coming to lab, you will lose 40% of the points for that lab.

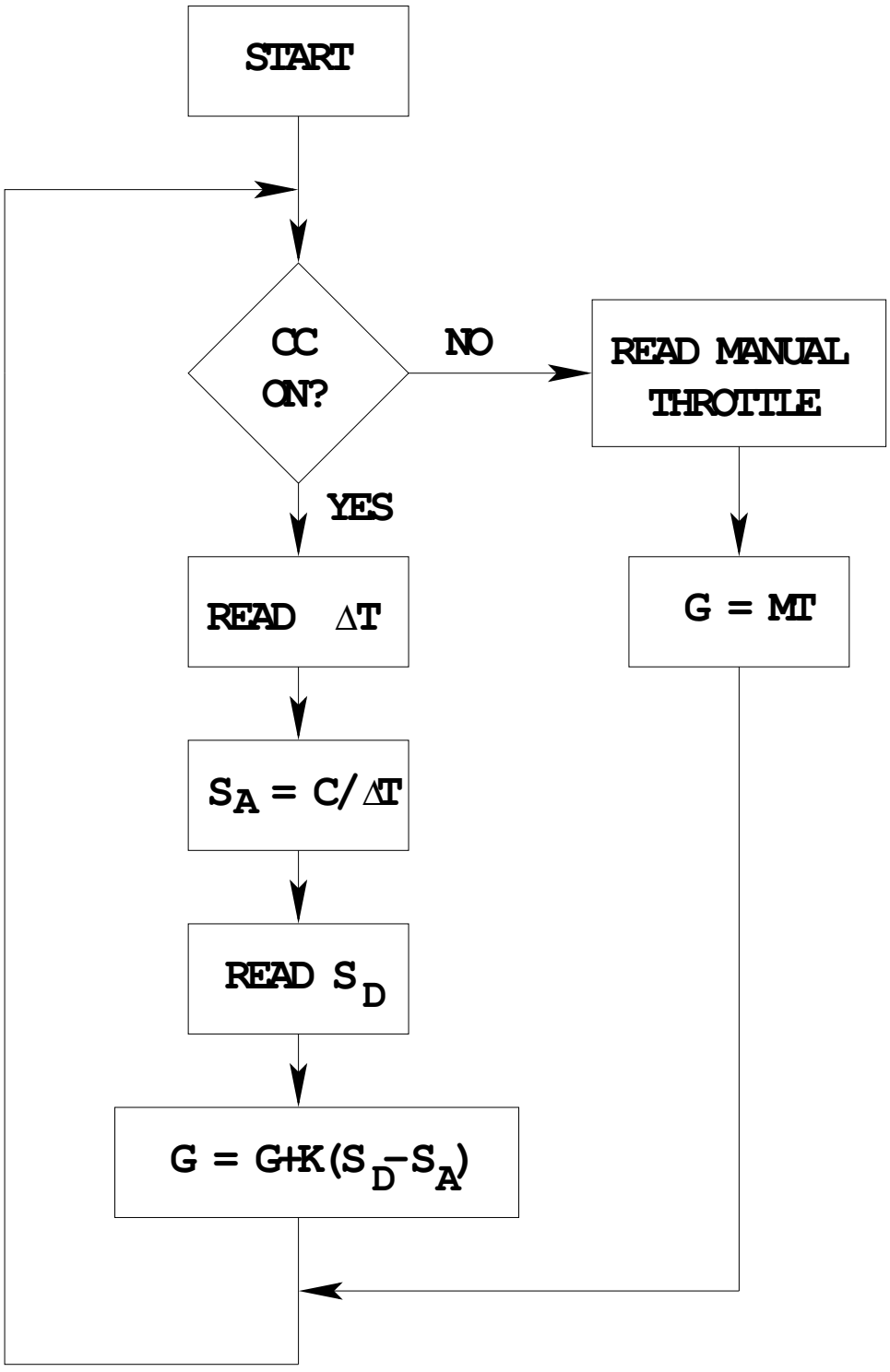

AUTOMOBILE CRUISE CONTROL
THROTTLE
CONTROL
MEASURE SPEED
SET DESIRED SPEED

AUTOMOBILE CRUISE CONTROL

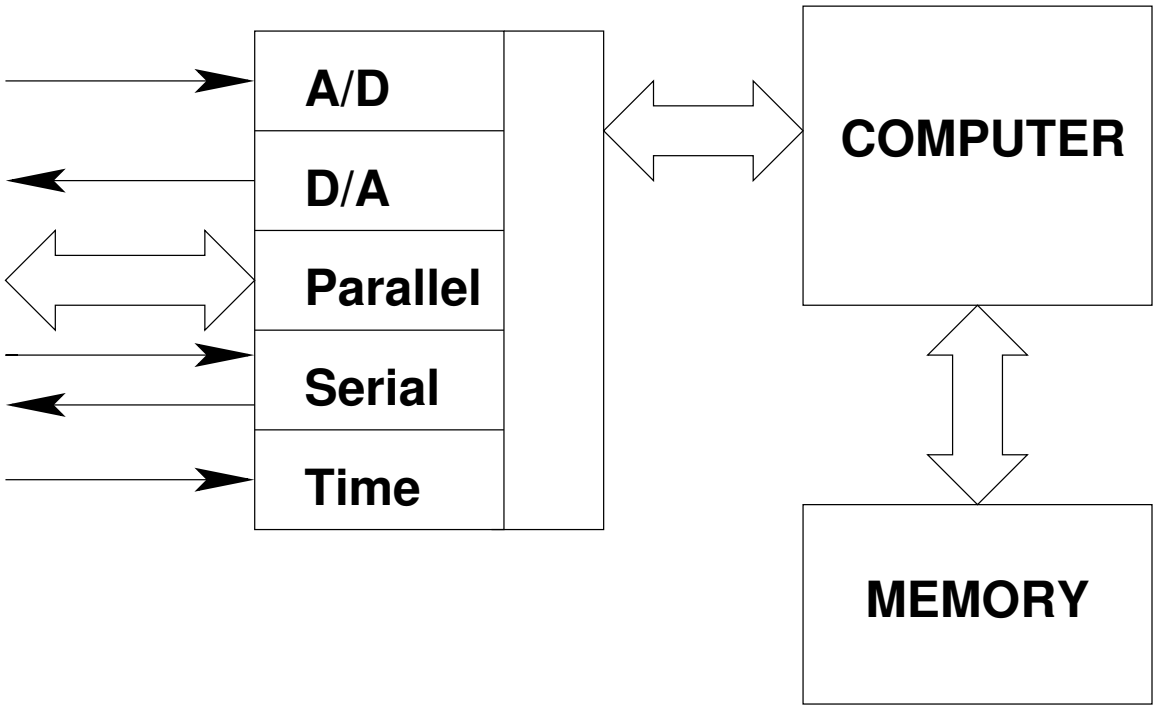


ΔT = time for one revolution of wheel.

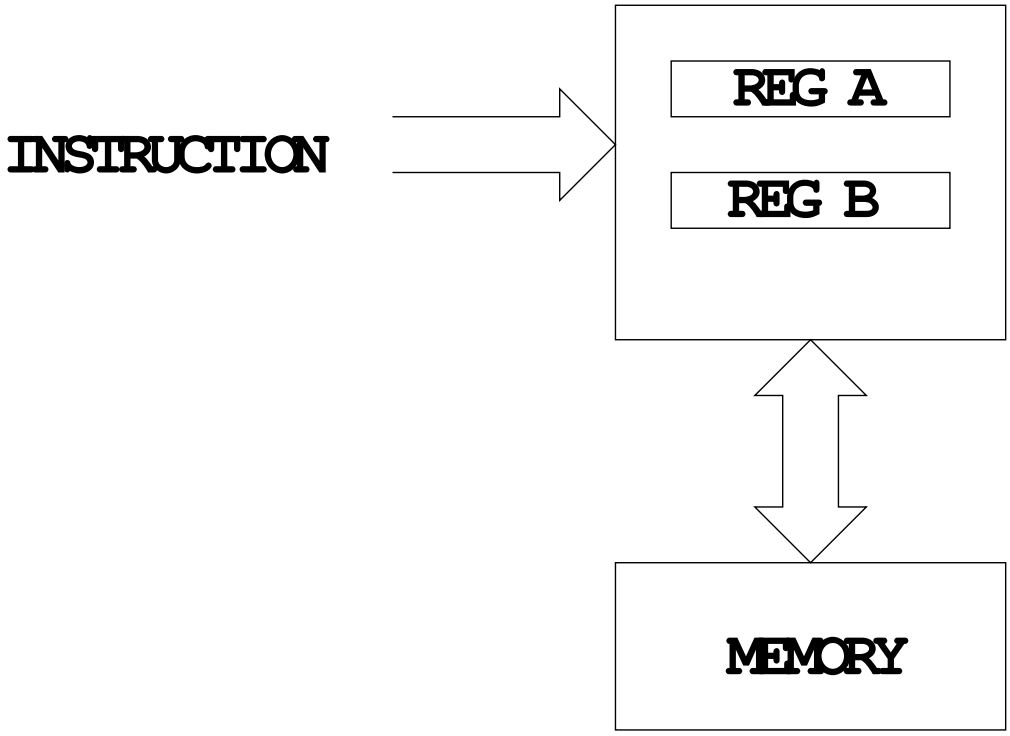
Speed = $C/\Delta T$, where C is the circumference of the wheel


START
CC ON?
NO
READ MANUAL THROTTLE
YES
READ ΔT
G = MT
$S_A = C/\Delta T$
READ S_D
$G = G+K(S_D-S_A)$

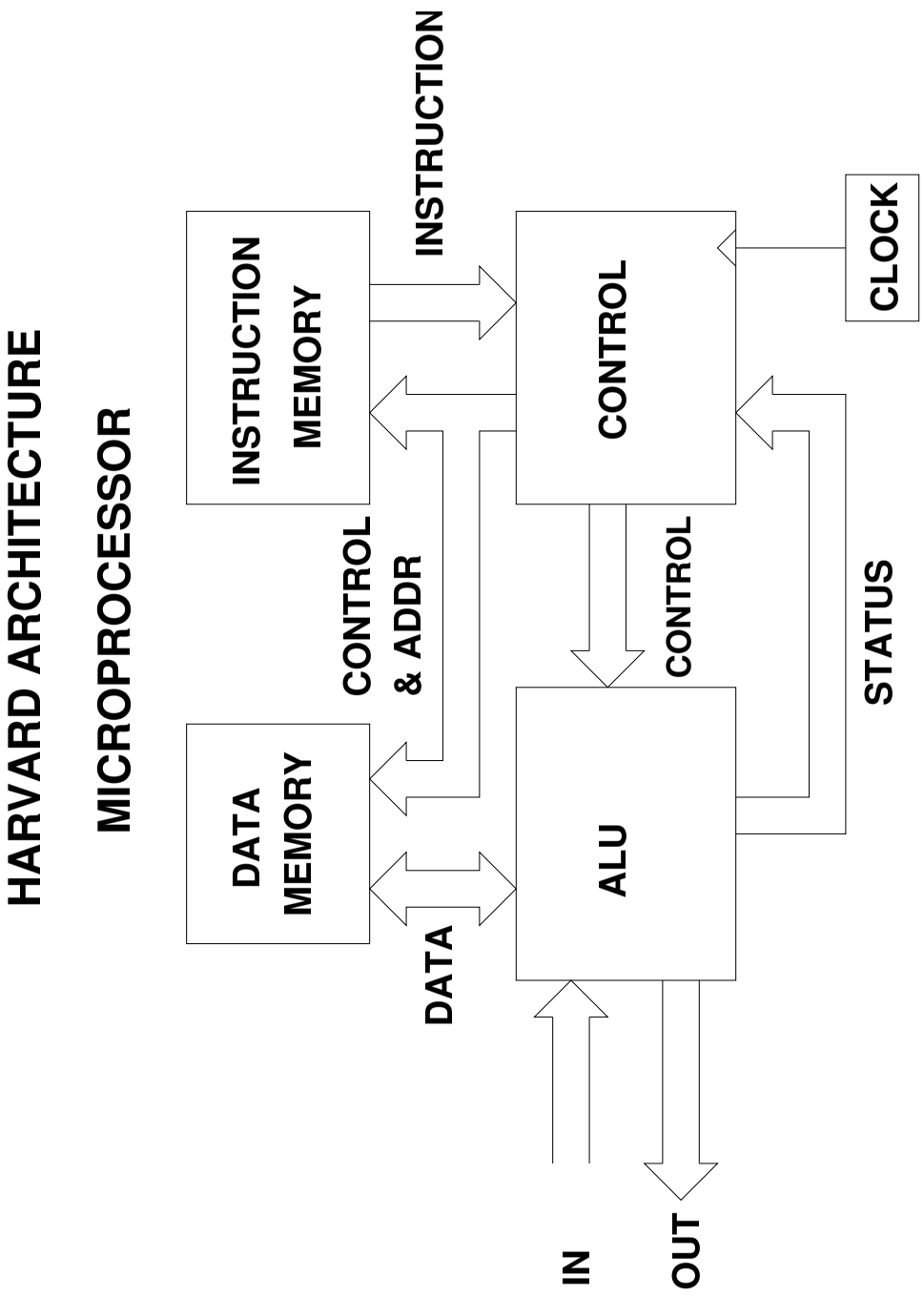
MICROCONTROLLER

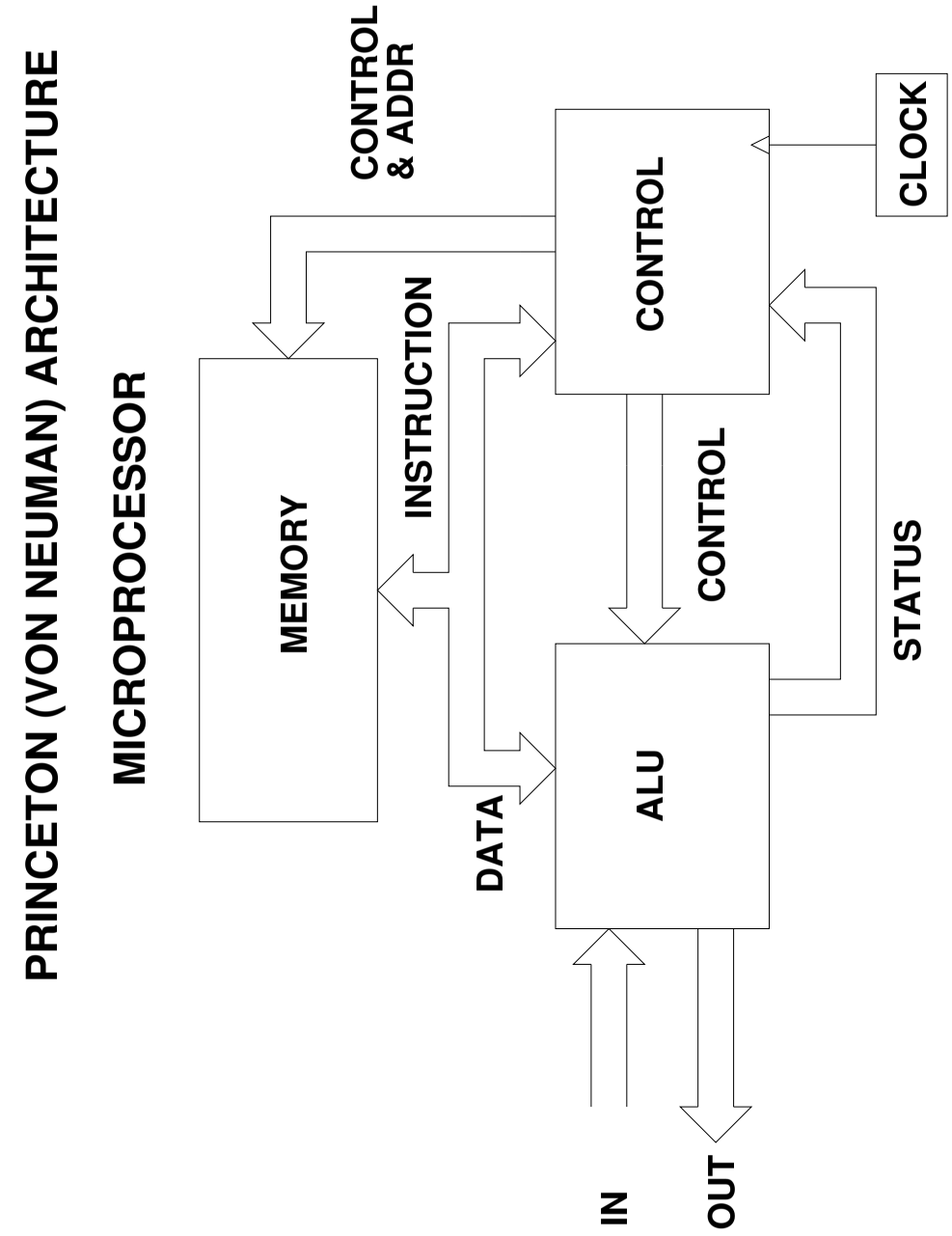


SIMPLE MICROPROCESSOR



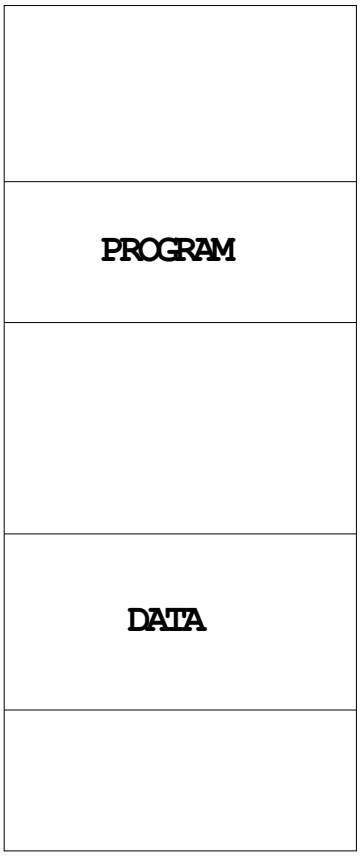
INSTRUCTION	ACTION
18 06	(A) + (B) => A
87	0 => A
5A 05	(A) => Address 5


HARVARD ARCHITECTURE
MICROPROCESSOR
INSTRUCTION MEMORY
INSTRUCTION
CONTROL
CLOCK
CONTROL & ADDR
CONTROL
STATUS
DATA MEMORY
ALU
DATA
IN
OUT

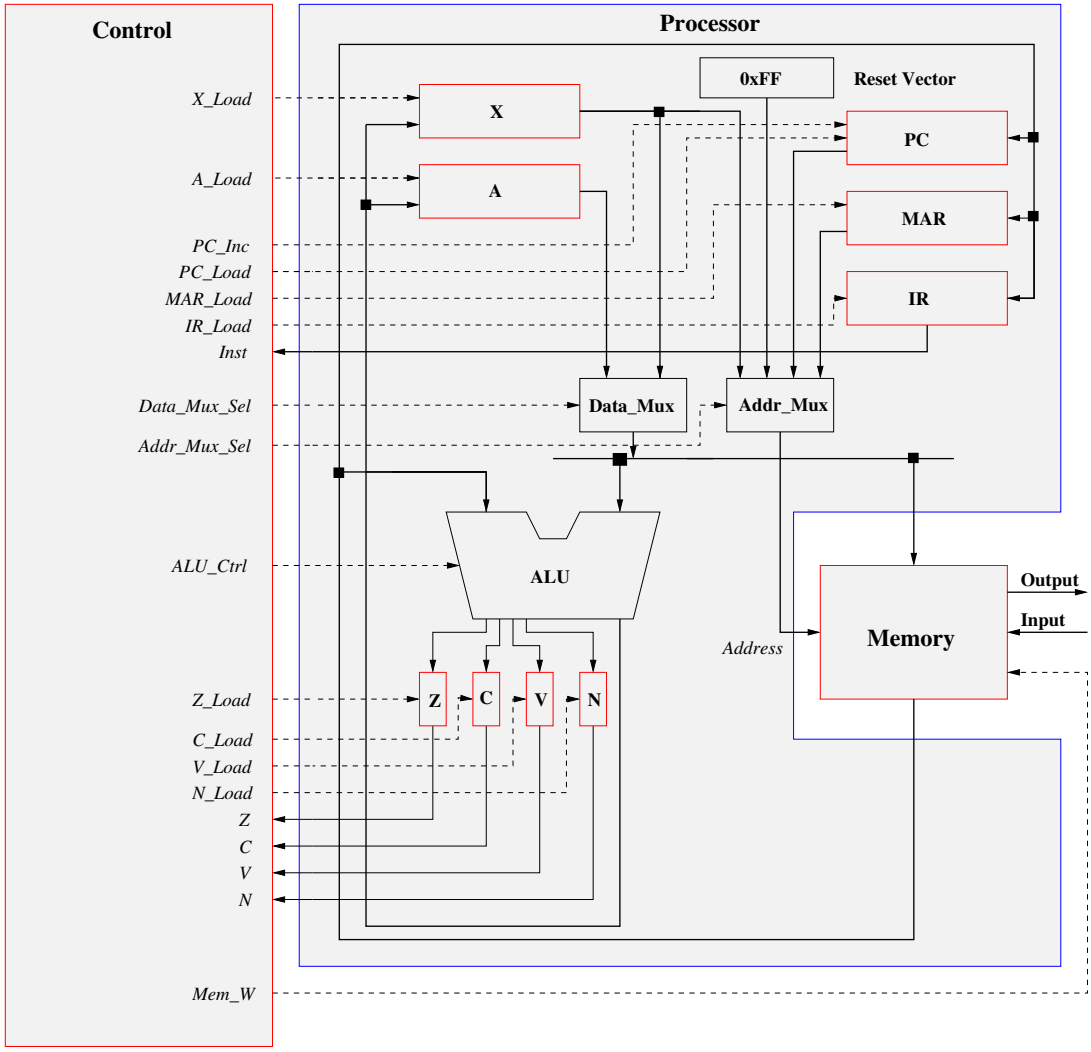

PRINCETON (VON NEUMAN) ARCHITECTURE
MICROPROCESSOR
CONTROL & ADDR
CONTROL
CLOCK
MEMORY
INSTRUCTION
CONTROL
STATUS
DATA
ALU
IN
OUT

MEMORY MAP

(Princeton Architecture)



Function of memory determined by programmer


Control
Processor
X_Load
A_Load
PC_Inc
PC_Load
MAR_Load
IR_Load
Inst
Data_Mux_Sel
Addr_Mux_Sel
ALU_Ctrl
Z_Load
C_Load
V_Load
N_Load
Z
C
V
N
Mem_W
X
A
0xFF
Reset Vector
PC
MAR
IR
Data_Mux
Addr_Mux
ALU
Z
C
V
N
Address
Memory
Output
Input

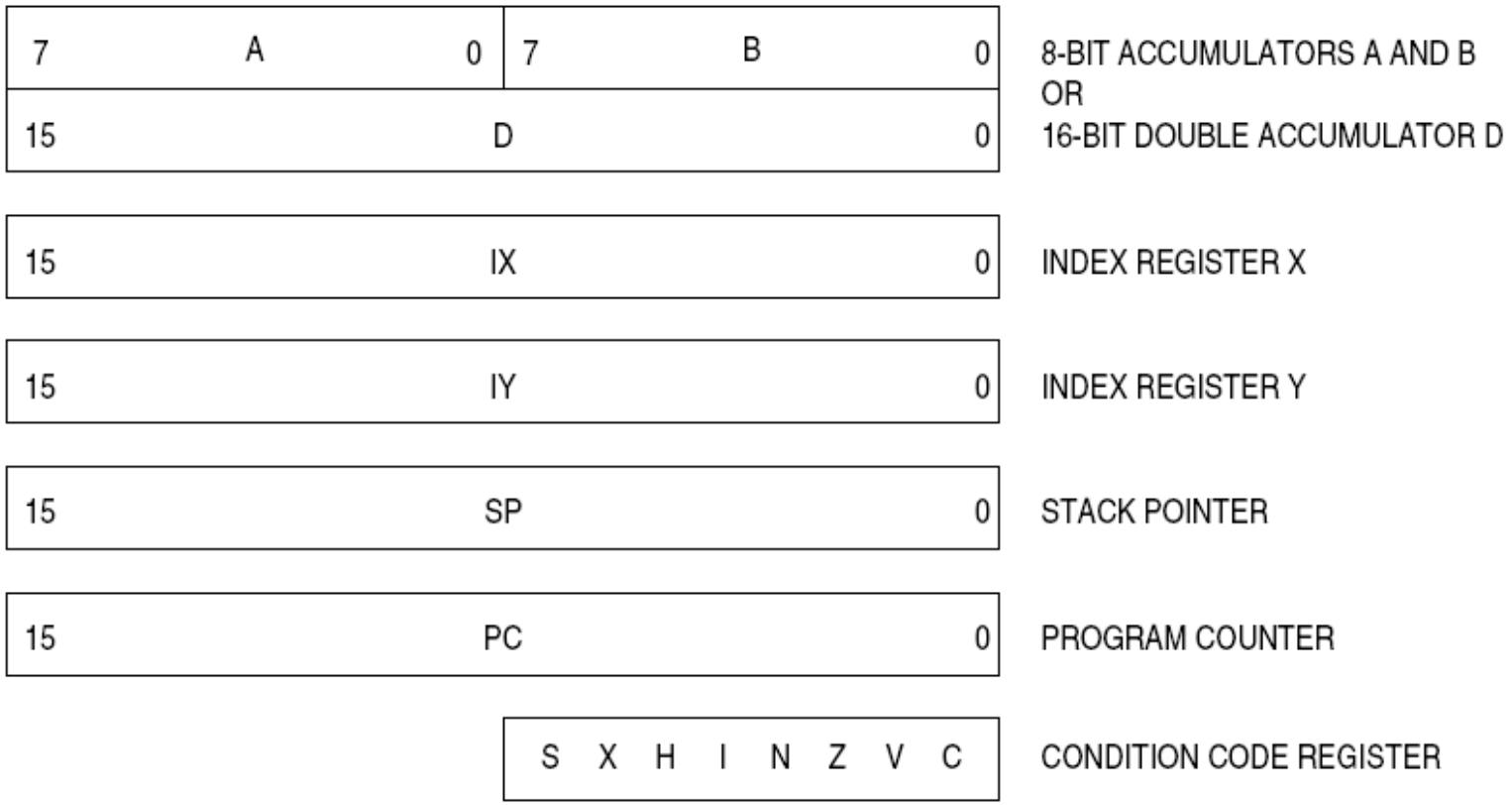


Figure 2-1. Programming Model