CpE 213
Digital Systems Design
WIMP51

Lecture 8

Friday 9/9/2005

UMR UNIVERSITY OF MISSOURI-ROLLA

The Name. The Degree. The Difference.

Overview

- Announcements
- WIMP51

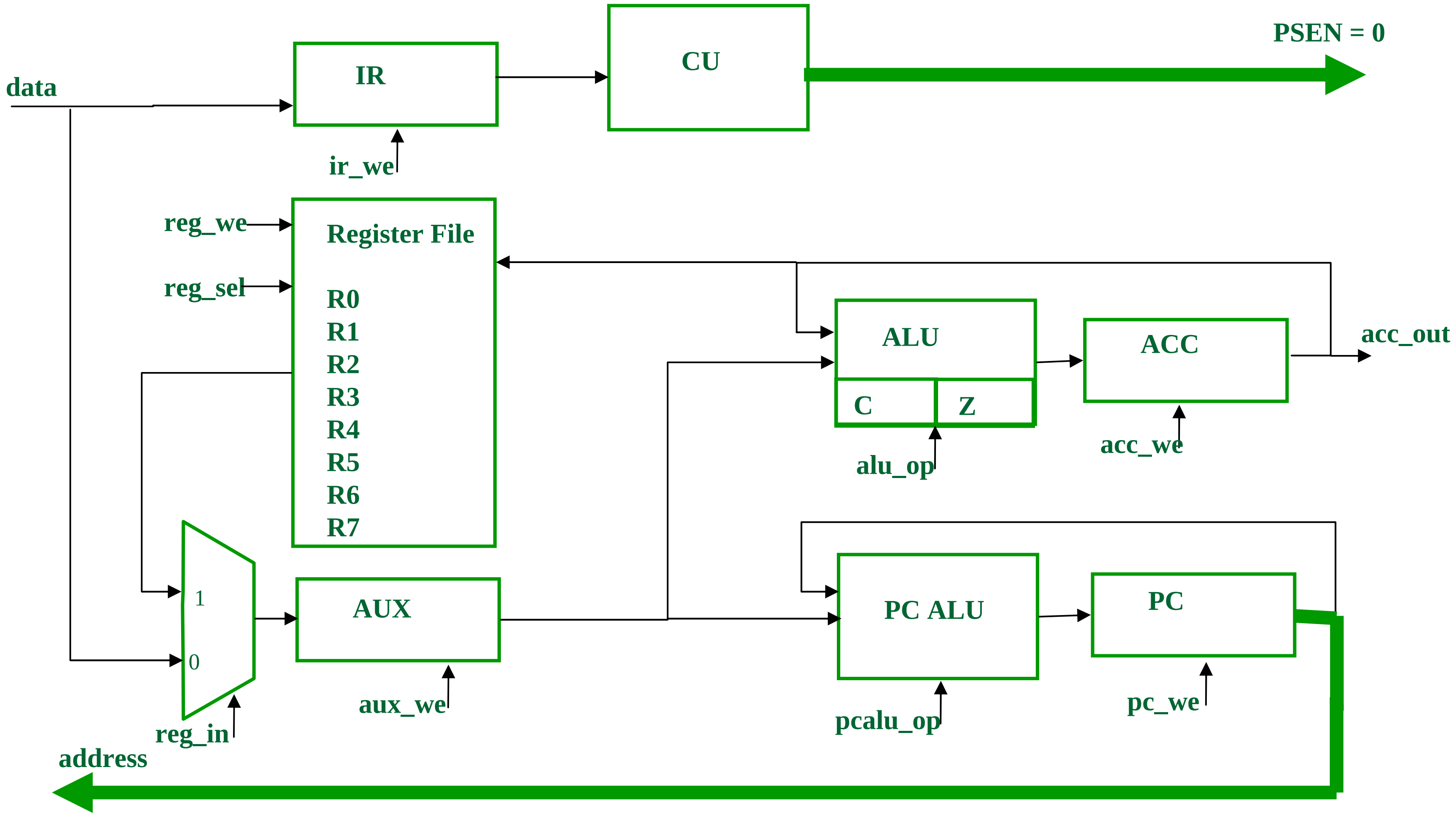
Announcements

- Homework 3 is due on Friday.
- Interesting URL:
  - http://microcontroller.com/EmbeddedSystems.asp?c=4

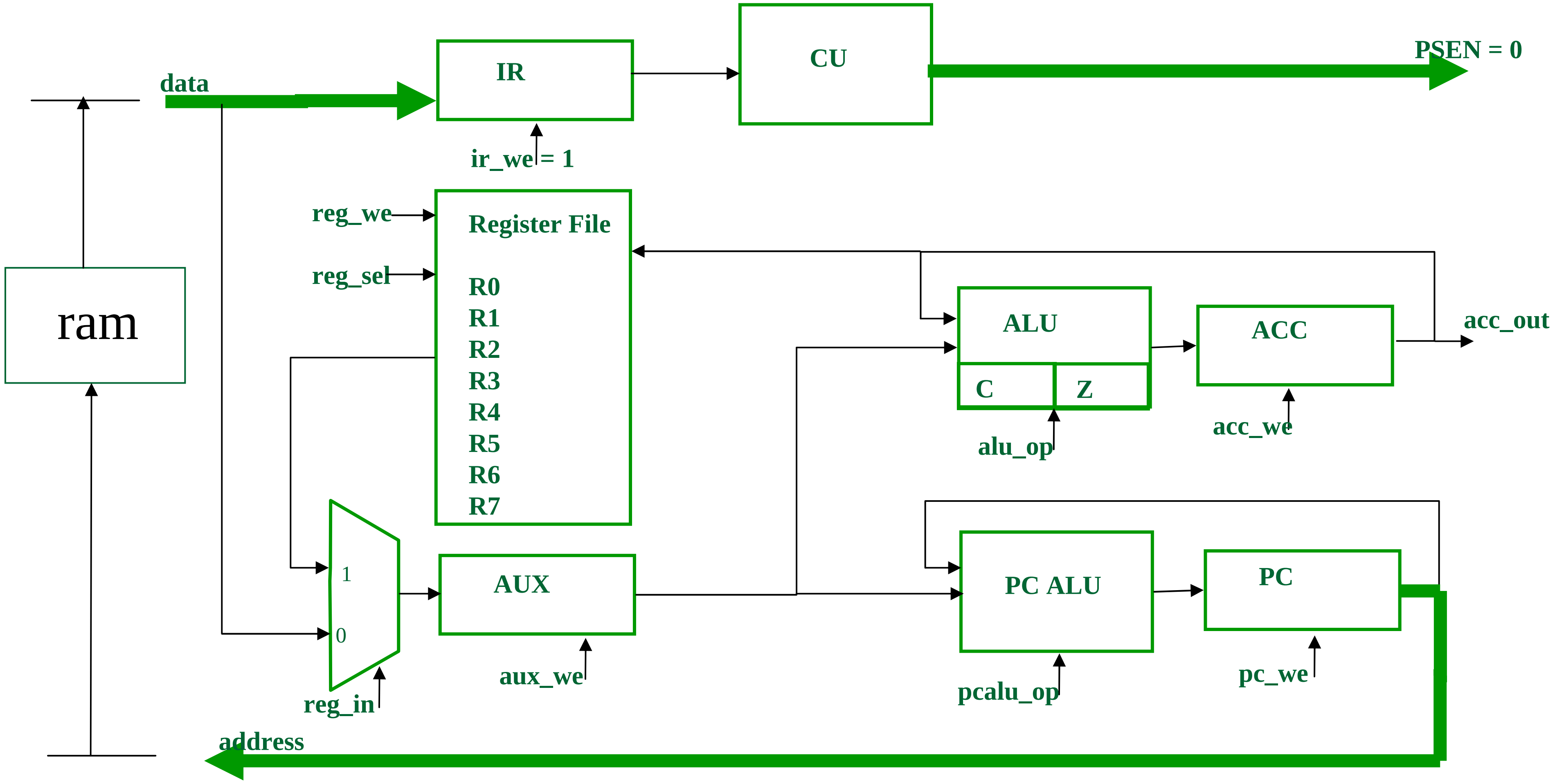
WIMP51
Weekend Instructional Microprocessor

Some slides adapted from Dr. Hardy J. Pottinger

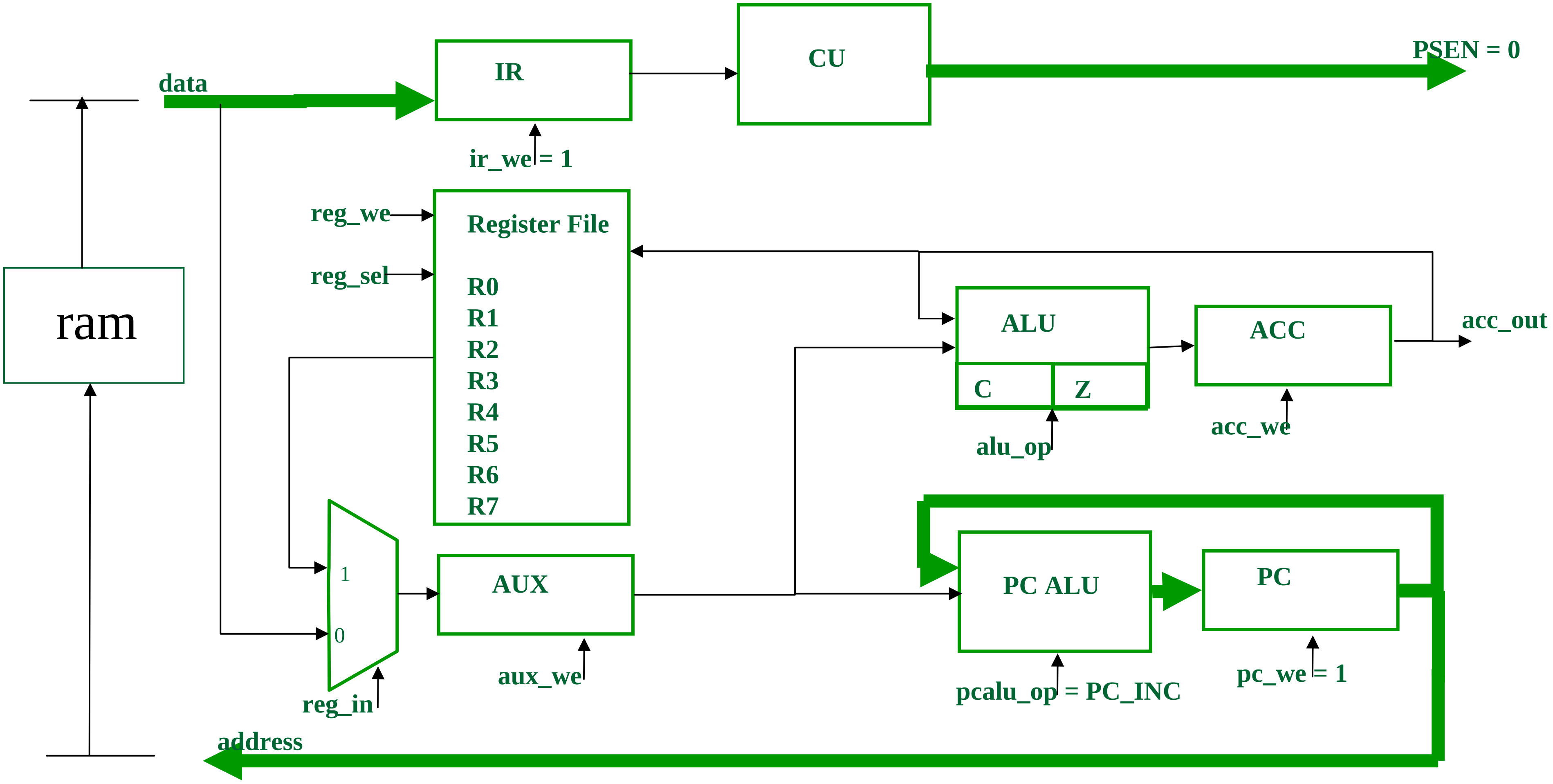
Fetch cycle step 1: addr <= PC



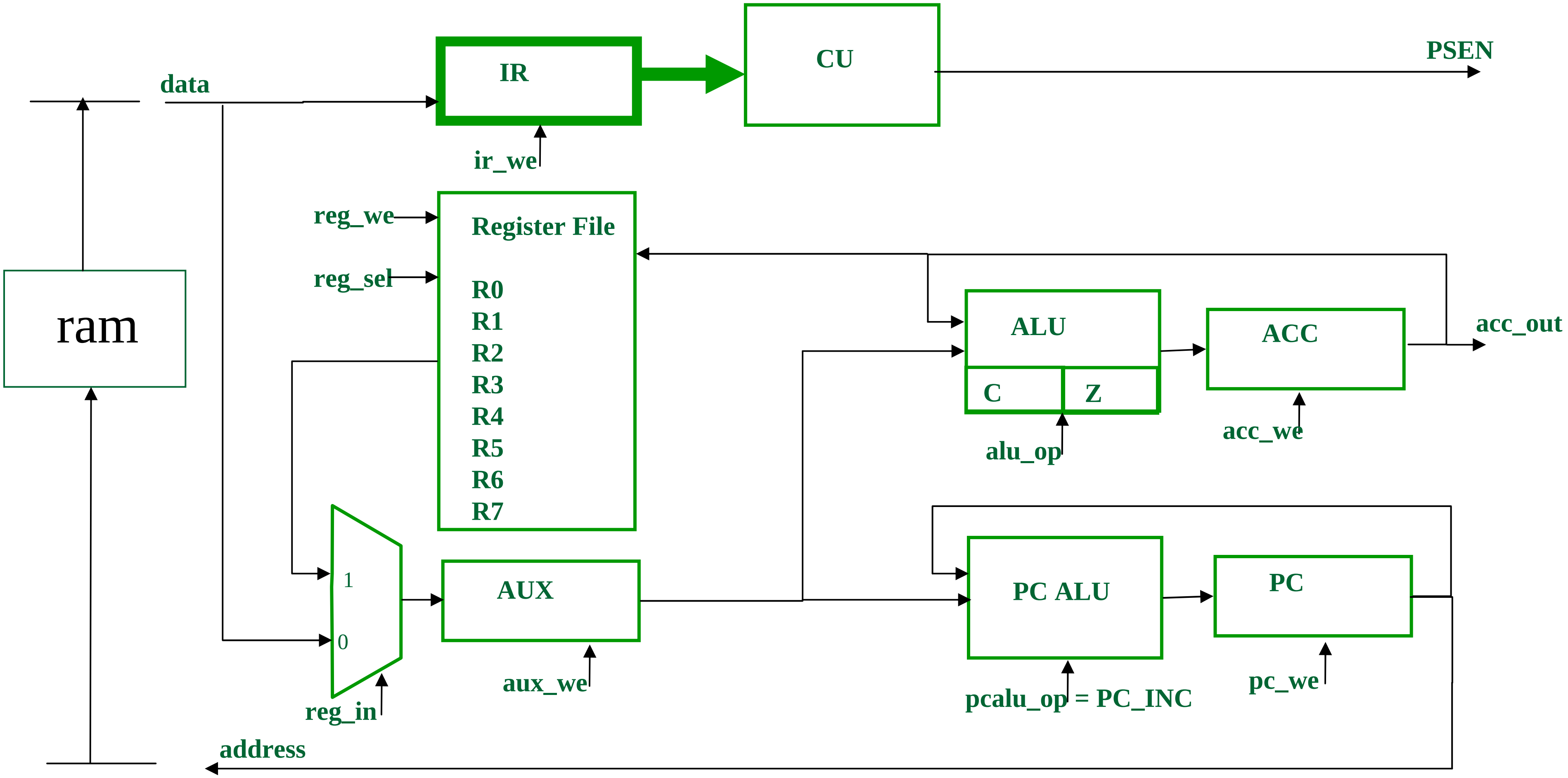
Fetch cycle step 2: IR<= data



Fetch cycle: PC<= PC+1



Decode cycle: decode current instruction in IR

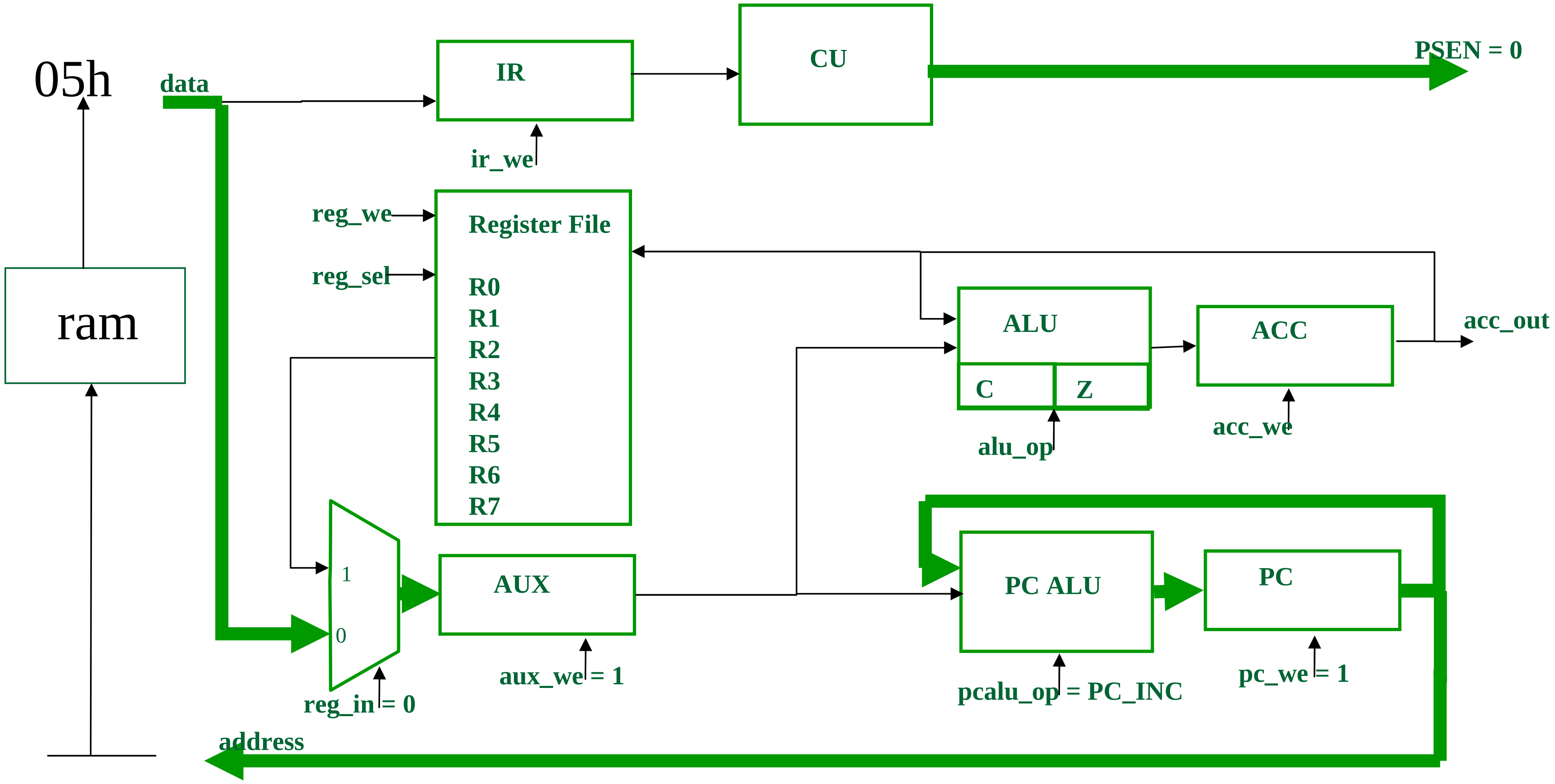


Wimp51 Instructions

MOV A, #D	01110100	dddddddd	A<=D
ADDC A, #D	00110100	dddddddd	C,A<=A+D+C
MOV Rn, A	11111nnn		Rn<=A
MOV A, Rn	11101nnn		A<=Rn
ADDC A, Rn	00111nnn		C,A<=A+Rn+C
ORL A, Rn	01001nnn		A<=A OR Rn
ANL A, Rn	01011nnn		A<=A AND Rn
XRL A, Rn	01101nnn		A<=A XOR Rn
SWAP A	11000100		A<=$A_{(3-0)}$ SWAP $A_{(7-4)}$
CLR C	11000011		C<=0
SETB C	11010011		C<=1
SJMP rel	10000000	aaaaaaaa	PC<=PC+rel+2
JZ rel	01100000	aaaaaaaa	PC<=PC+rel+2 if Z

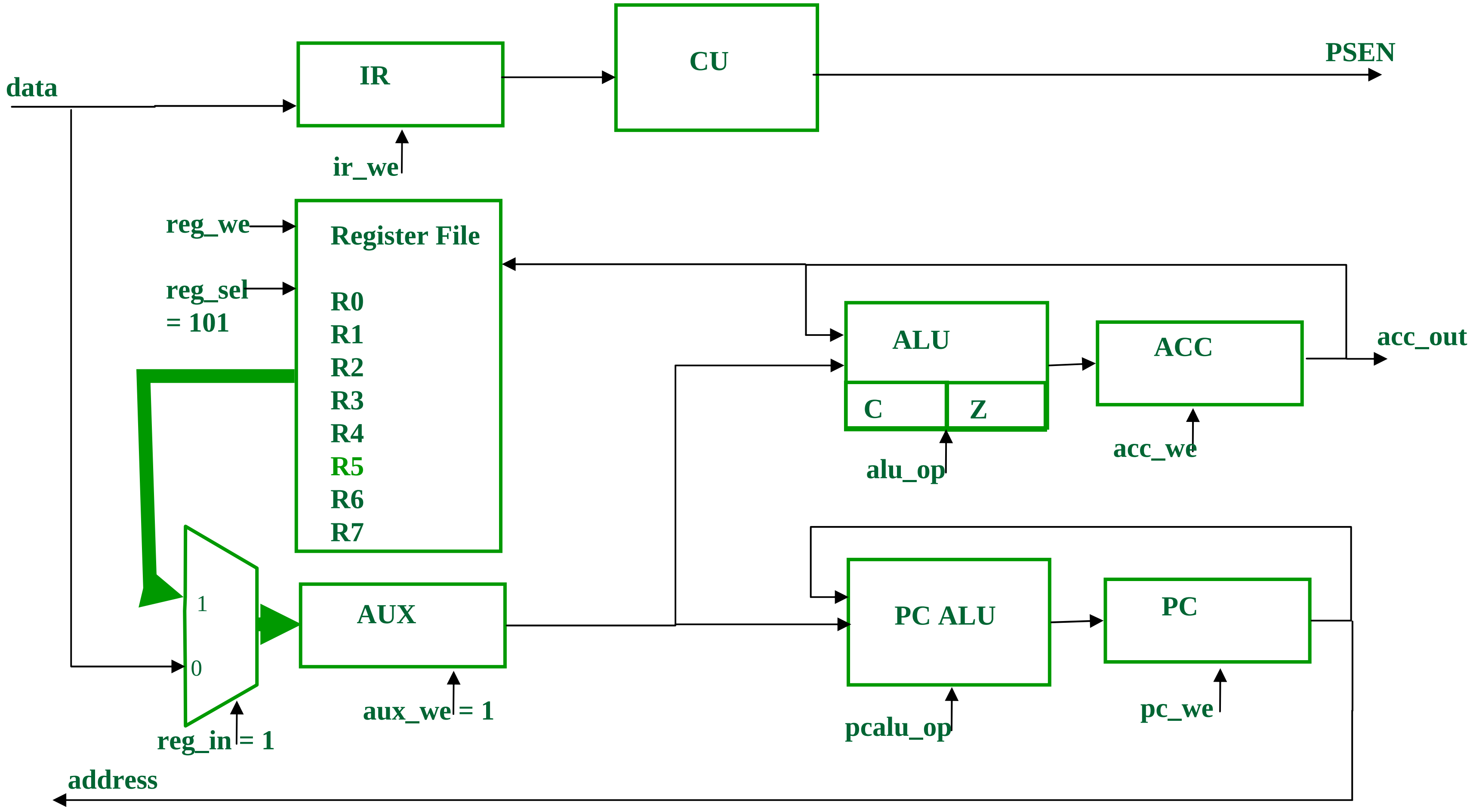
Decode cycle: immediate source

```
MOV A, #05h
```



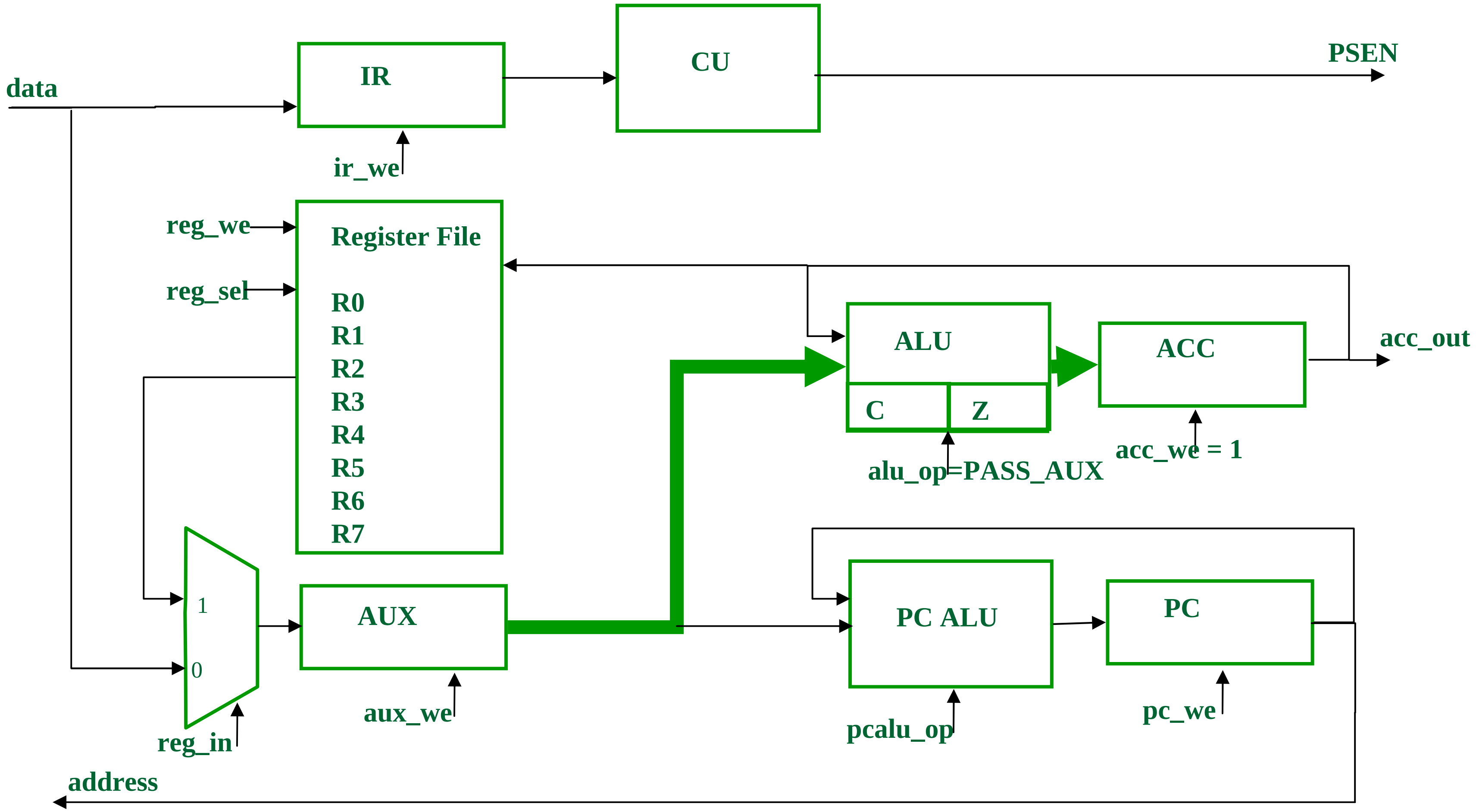
Decode cycle: register source

MOV A, R5



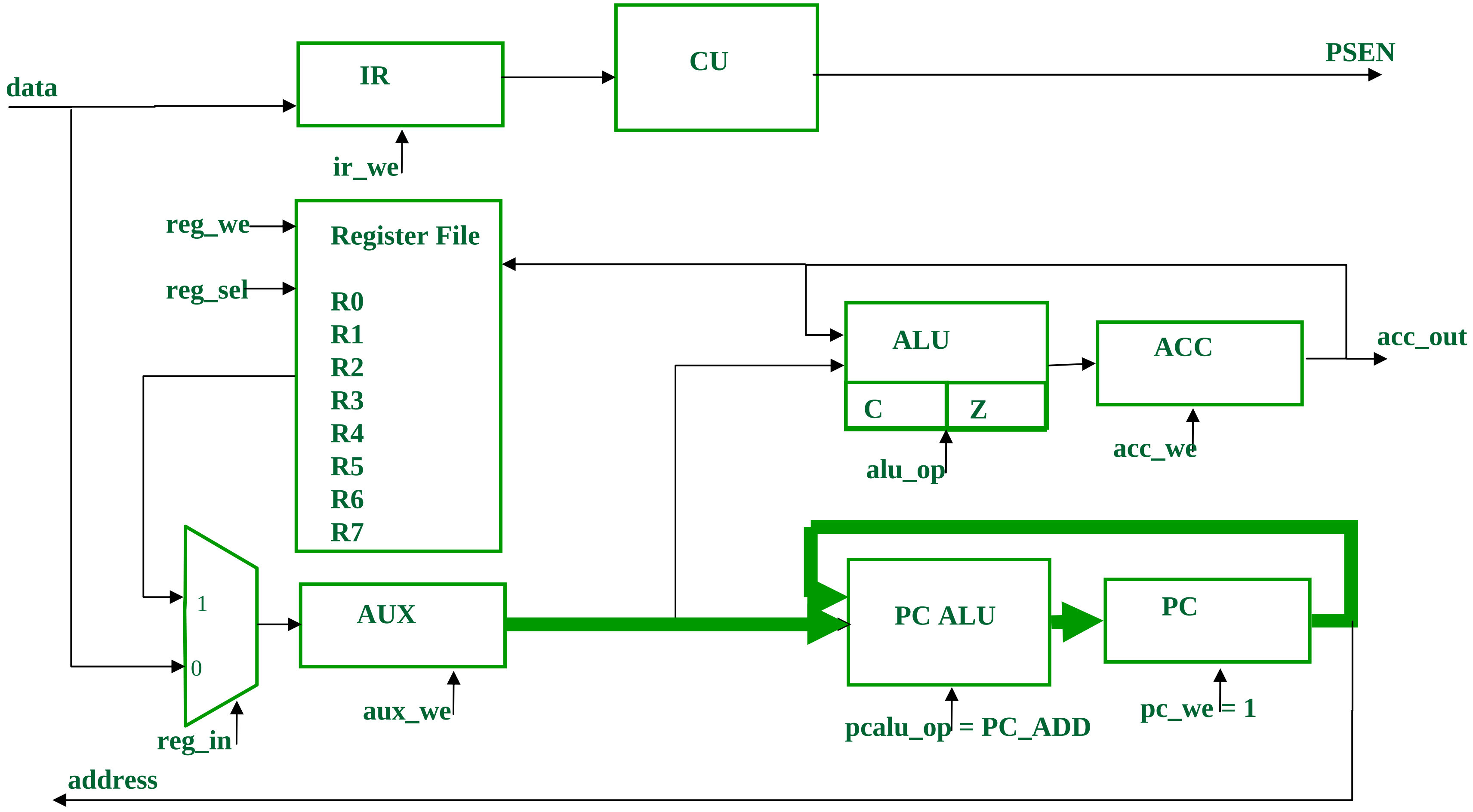
Execute cycle

MOV A, R5



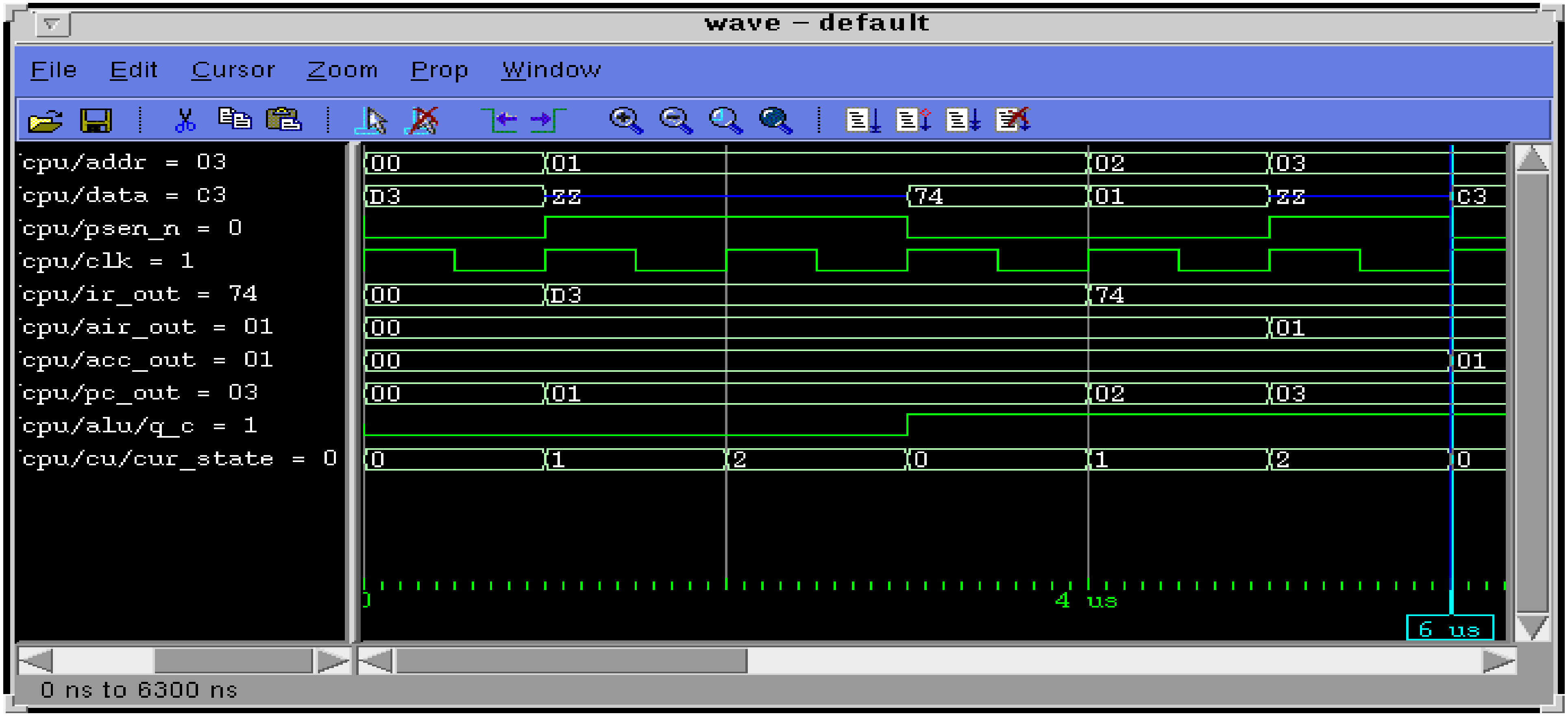
Execute cycle

```
SJMP loop
```



WIMP51 timing diagram

- Two complete instruction cycles are shown.
- First is a SETB C (D3h) at location 0000.
- PC increments at end of Fetch cycle, C set at end of Exec cycle.
- What is the second instruction? What is the value of the data bus during second fetch?

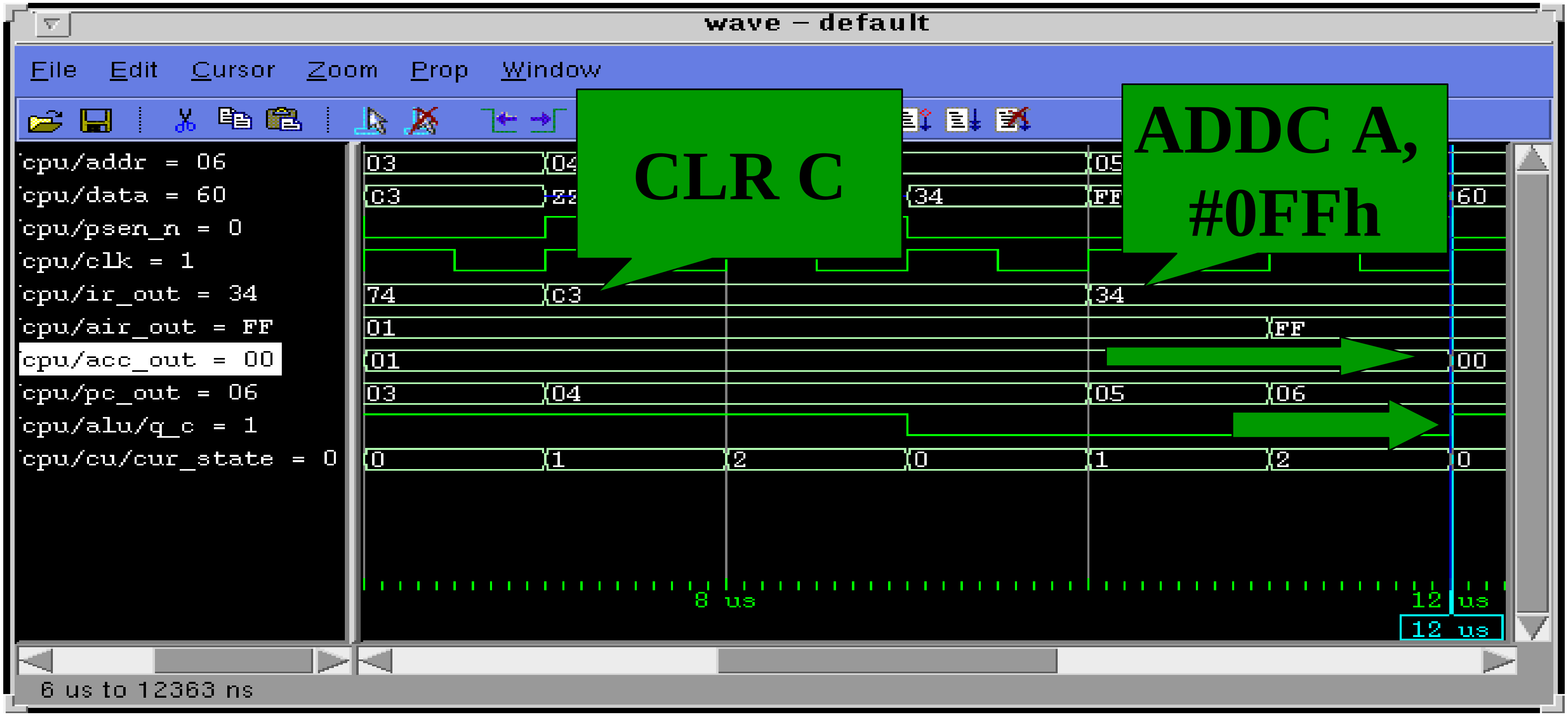


```
cpu/addr = 03
cpu/data = C3
cpu/psen_n = 0
cpu/clk = 1
cpu/ir_out = 74
cpu/air_out = 01
cpu/acc_out = 01
cpu/pc_out = 03
cpu/alu/q_c = 1
cpu/cu/cur_state = 0
```

0 ns to 6300 ns

WIMP51 timing diagram 2

- Two more instructions. What are they?
- Note ACC goes from 1 to 0 at end of second instruction.
- C is set at end of second instruction.



cpu/addr = 06
cpu/data = 60
cpu/psen_n = 0
cpu/clk = 1
cpu/ir_out = 34
cpu/air_out = FF
cpu/acc_out = 00
cpu/pc_out = 06
cpu/alu/q_c = 1
cpu/cu/cur_state = 0

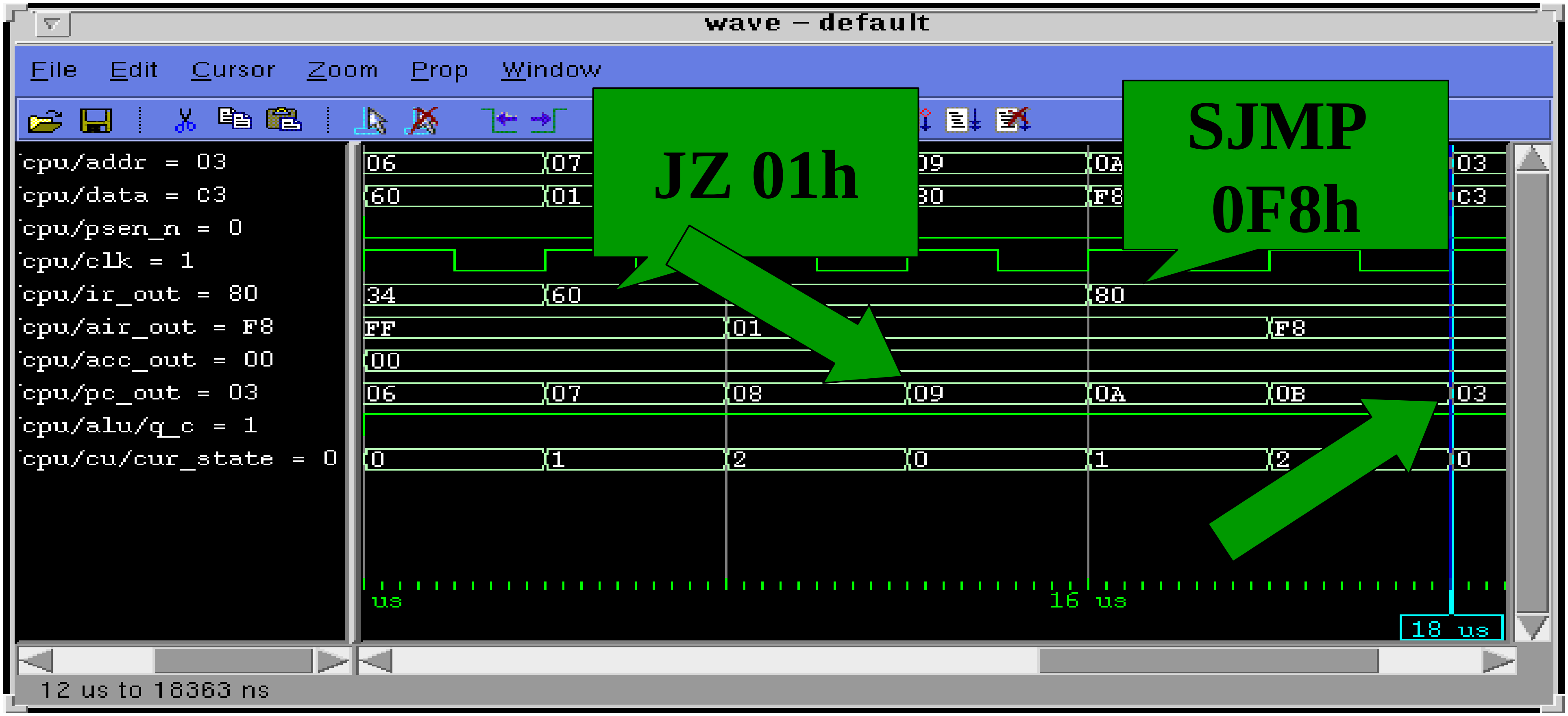
6 us to 12363 ns

CLR C

ADDC A, #0FFh

WIMP51 timing diagram 3

- Two more instructions. What are they?
- First instruction increments PC once more.
- Second instruction increments PC then changes it to 3H.



```
cpu/addr = 03
cpu/data = C3
cpu/psen_n = 0
cpu/clk = 1
cpu/ir_out = 80
cpu/air_out = F8
cpu/acc_out = 00
cpu/pc_out = 03
cpu/alu/q_c = 1
cpu/cu/cur_state = 0
```

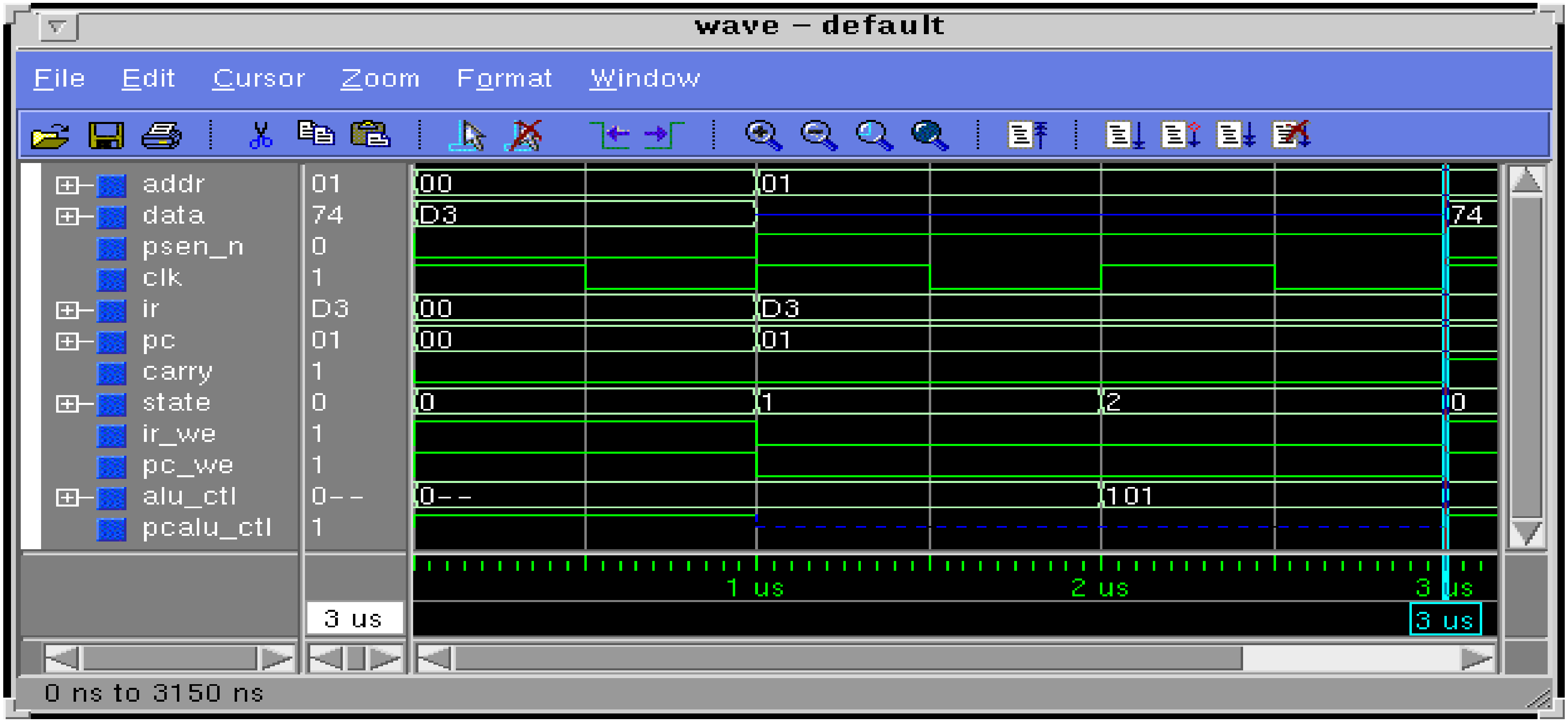
12 us to 18363 ns

JZ 01h

SJMP 0F8h

WIMP51 timing diagram 4

- Examine the control signals for SETB C.
- WE signals let registers be updated on rising clock edge.
- Dashed line represents “don’t care” signals.



Wimp51 timing

- Registers get loaded by rising edge at end of cycle.
- Each instruction lasts three clock cycles.
- Each instruction cycle consists of a Fetch, Decode, and an Execute clock cycle.
- Classic 8051 is similar but has more options; will be more complex (12 or more clock cycles per instruction).

Wimp51 timing

- Timing diagrams generated by a simulator.
- Simulator ‘executes’ a model of WIMP51.
- Model is written in VHDL.
- VHDL can be used to synthesize WIMP51 hardware.
- Simulation models let us ‘try out’ a design before committing to hardware.
- We can simulate hardware, software, or both.
- Used for rapid prototyping and to get it right the first time.

An example: Calculate 2+1

```
Code                Addr          Instr
-----------------------------------------------
74 01               0000h         MOV  A,#01H
F8                  0002h         MOV  R0,A
74 02               0003h         MOV  A,#02H
C3                  0005h         CLR  C
38                  0006h         ADDC A,R0
F9                  0007h         MOV  R1,A
80 FE               0008h  Stop:  SJMP Stop
```

An example: Calculate 2+1

PC	R0	R1	Acc	C		Instruction
0	??	??	??	?		MOV A,#1
2	??	??	1	?		MOV R0,A
3	1	??	1	?		MOV A,#2
5	1	??	2	?		CLR C
6	1	??	2	0		ADDC A,R0
7	1	??	3	0		MOV R1,A
8	1	3	3	0	Stop:	SJMP Stop

Calculate 3*5 with for loop

```
Code      Addr       Instr
                     MOV  A,#0
                     MOV  R0,A
                     MOV  A,#3
                     MOV  R1,A
          Loop:      MOV  A,#5
                     CLR  C
                     ADDC A,R0
                     MOV  R0,A
                     MOV  A,R1
                     CLR  C
                     ADDC A,#0FFh
                     MOV  R1,A
                     JZ   Stop
                     SJMP Loop
          Stop:      SJMP Stop
```

For Monday

- Review today’s lecture notes and textbook.
- Download and read WIMP51 handout.
- Begin Assignment 3.