CpE 213
Digital Systems Design
Computer Organization and WIMP51

Lecture 6

Friday 9/2/2005

UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.

Overview

- Announcements
- Introduction to computer organization
- WIMP51

Announcements

- Homework 2 is due on Friday.
- Interesting URL:
  - http://microcontroller.com/EmbeddedSystems.asp?c=4

Detour: WIMP51 Program

Wimp51 Instructions

```
MOV A, #D  01110100 dddddddd
ADDC A, #D 00110100 dddddddd
MOV Rn, A  11111nnn
MOV A, Rn  11101nnn
ADDC A, Rn 00111nnn
ORL A, Rn  01001nnn
ANL A, Rn  01011nnn
XRL A, Rn  01101nnn
SWAP A     11000100
CLR C      11000011
SETB C     11010011
SJMP rel   10000000 aaaaaaaa
JZ rel     01100000 aaaaaaaa
```

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

Introduction to Computer Organization

Some slides adapted from Dr. Shoukat Ali

Some definitions

- program: a set of instructions
- computer: a device that sequentially executes a stored program
- microprocessor: major functional blocks of a computer packaged in SINGLE chip
- microcontroller: a microprocessor PLUS a number of peripherals INTEGRATED into a SINGLE chip
- computer architecture: the arrangement and interconnection of its functional blocks
- instruction set of a computer: the set of operations the computer can be programmed to perform on data

von Neumann machine

- functional blocks of von Neumann machine
  - a memory, containing instructions and data
  - a processing unit, for performing arithmetic and logical operations
  - a control unit, for interpreting instructions
- basic sequence of operations
  1. get the first program instruction from the memory, fetch
  2. figure out what the instruction requires, decode
  3. execute the instruction, execute
  4. get the next program instruction, fetch
  5. go to step 2
- cycle through: ***fetch, decode, execute***

Micro operations

- each step in the sequence of operations consists of many micro-operations
- example
  - get the first program instruction from memory

1. find out the address where the instruction is located
2. send that address to the memory chip
3. enable the output of the memory chip
4. memory chip responds by placing the instruction on its “door”; get that instruction and bring it into a special register called the “instruction register”

- program counter register: a register that contains the address of the next instruction that the computer should execute
  - at start, it contains the address of the first instruction of the program

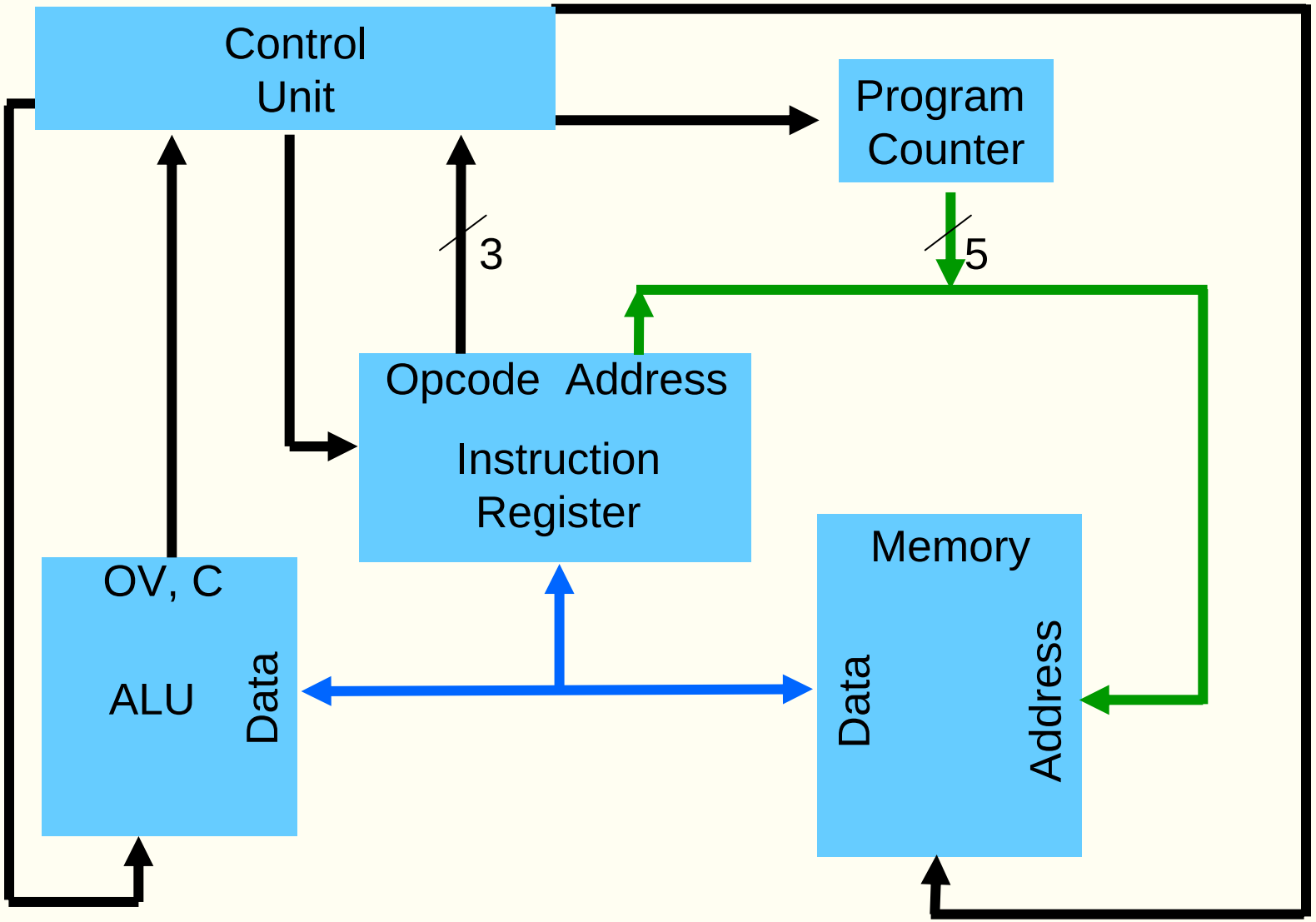
Diversion: programming model

- control unit uses two registers called the instruction register (IR) and the program counter (PC) to get the first (and subsequent) program instruction(s) from memory.
- the “user” or programmer cannot access instruction register.
- programming model of a computer: the set of registers available to a programmer
  - assume that for our simple computer, the user can access the PC, accumulator, and condition code register
    - condition code register: 2-bits, overflow and carry flags (simplistic)
- real life programming models
  - Alpha 21264 has 63 registers, each 64 bit wide
  - Intel Pentium 4 has 32 registers of width varying from 32 to 128 bits
  - Sun Sparc can have up to 520 registers!!!

Control unit

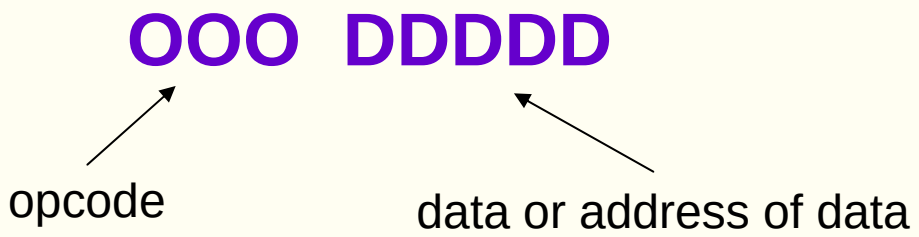
- orchestrates execution of the program
- consists of IR, PC, and sequential and combinational logic
- IR contains the current instruction
- PC contains the address of the next instruction to be executed
- control unit tasks
  - fetch: read an instruction from memory
    - the instruction's address is in the PC
  - decode: interpret the instruction, and then generate all those signals that tell the other components what to do to get the job done

Very simple view



Instruction format

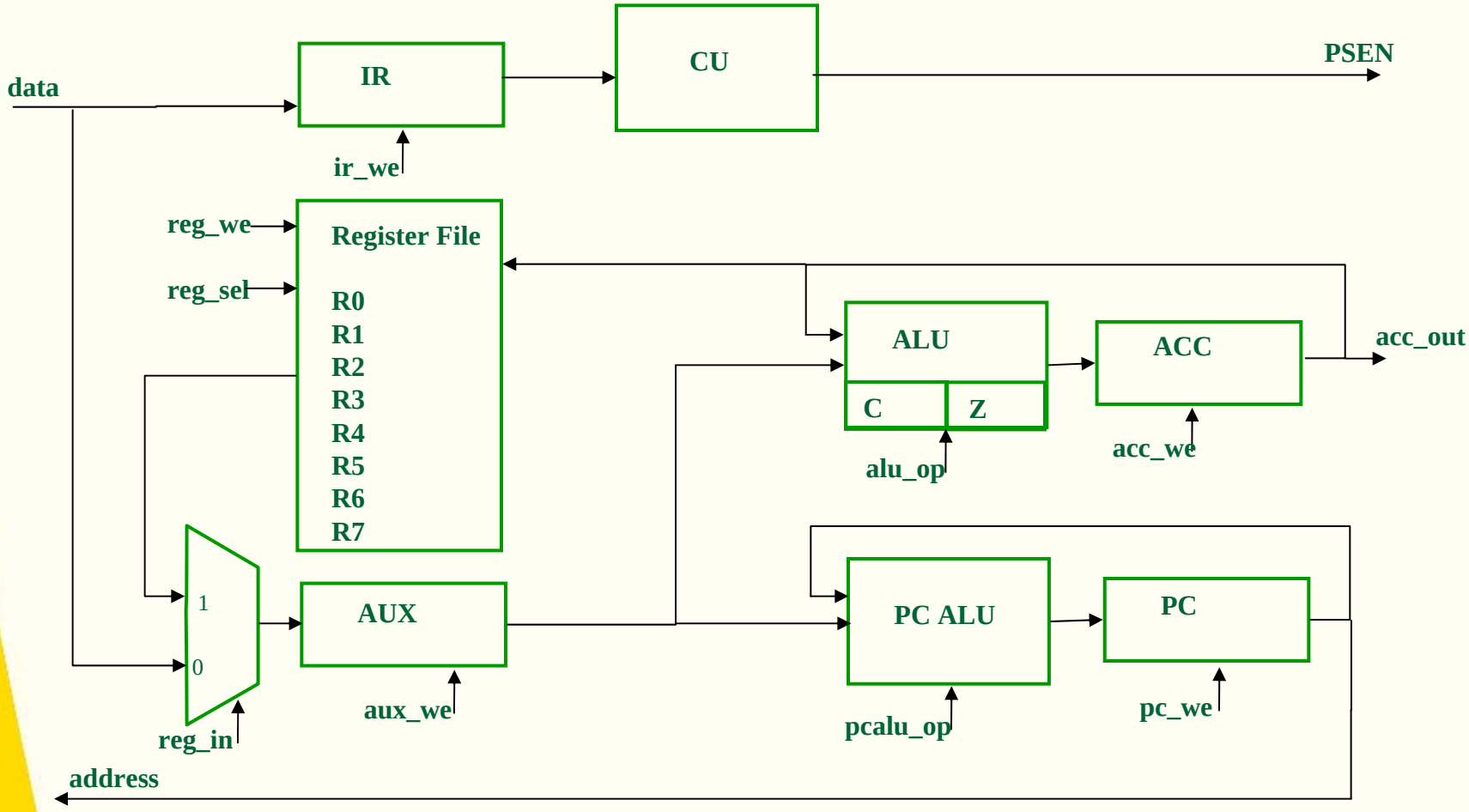
- for a general purpose computer, instructions typically consist of many fields
  - opcode field: indicates the operation to be performed
  - data field: contains either the data on which the operation is to be performed or the address in memory where the data can be found
- assume a simple “8-bit” computer that has
  - a 3-bit opcode
    - therefore allowing at most eight instructions in its instruction set
  - a 5-bit address field
    - thus allowing access to 32 locations



WIMP51
Weekend Instructional Microprocessor

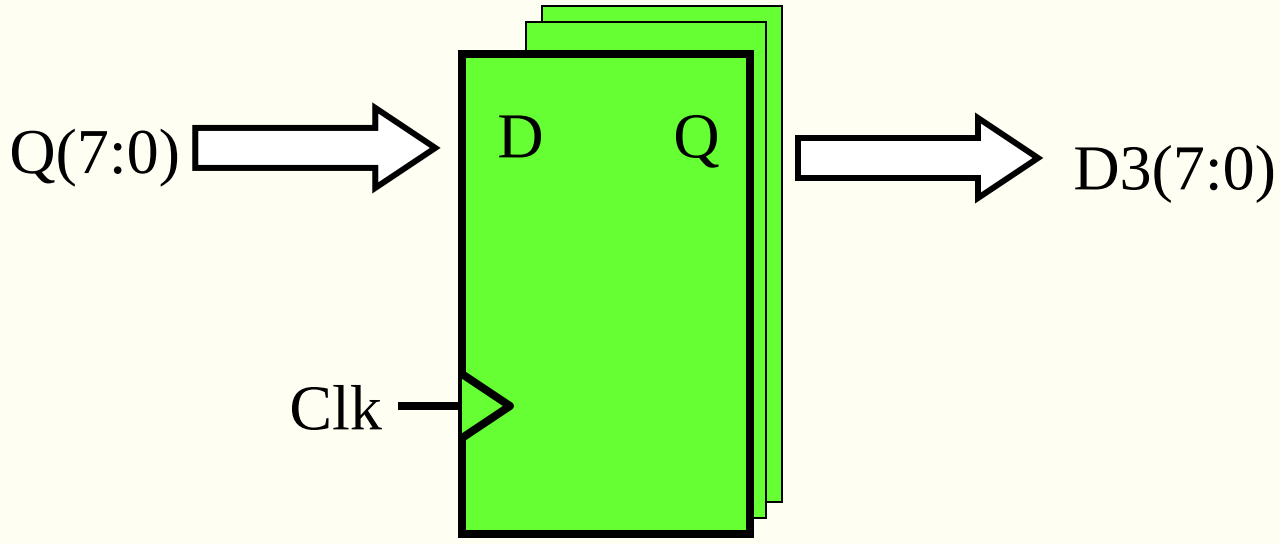
Some slides adapted from Dr. Hardy J. Pottinger

WIMP51 microprocessor

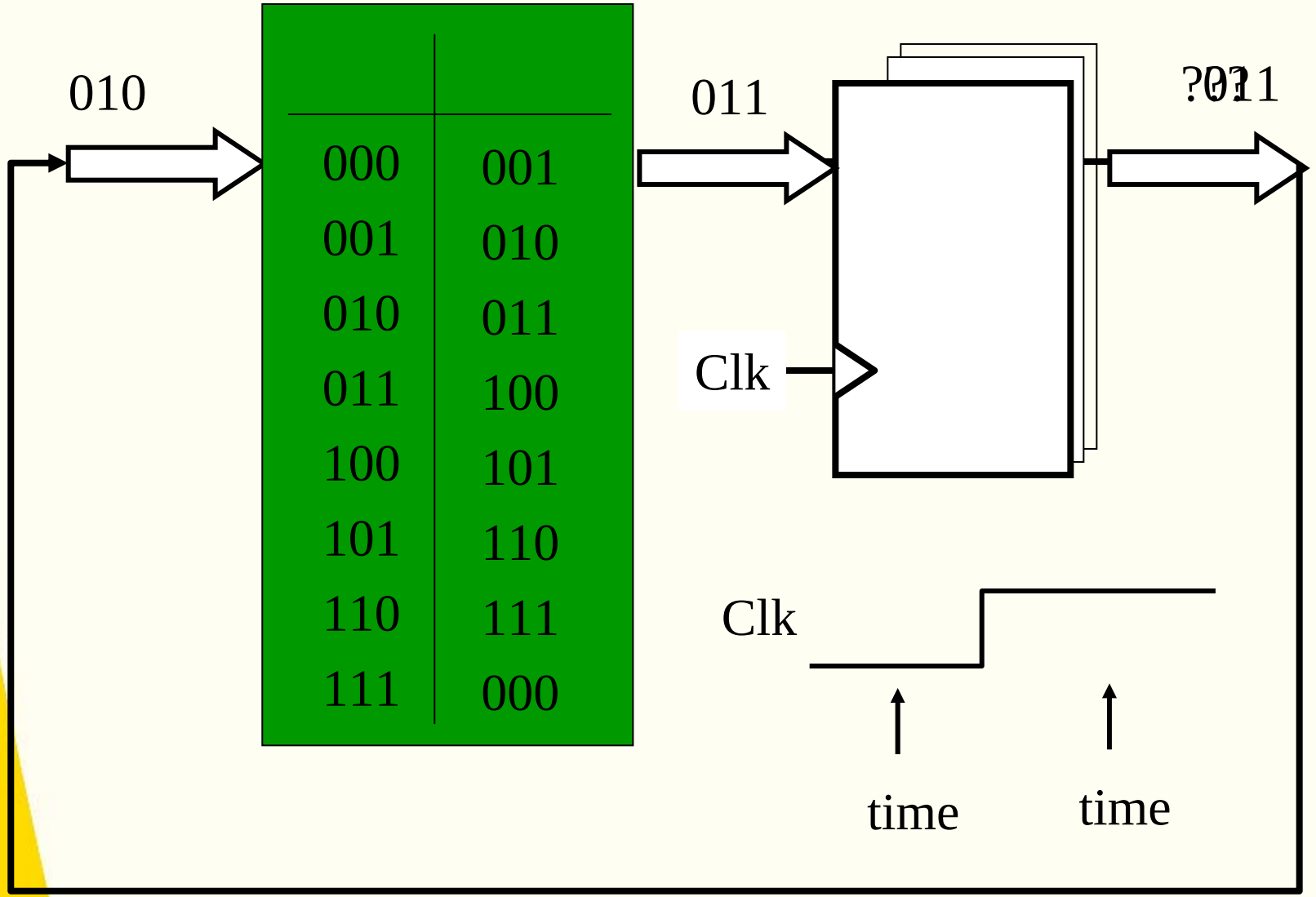


Register = array of flip-flops

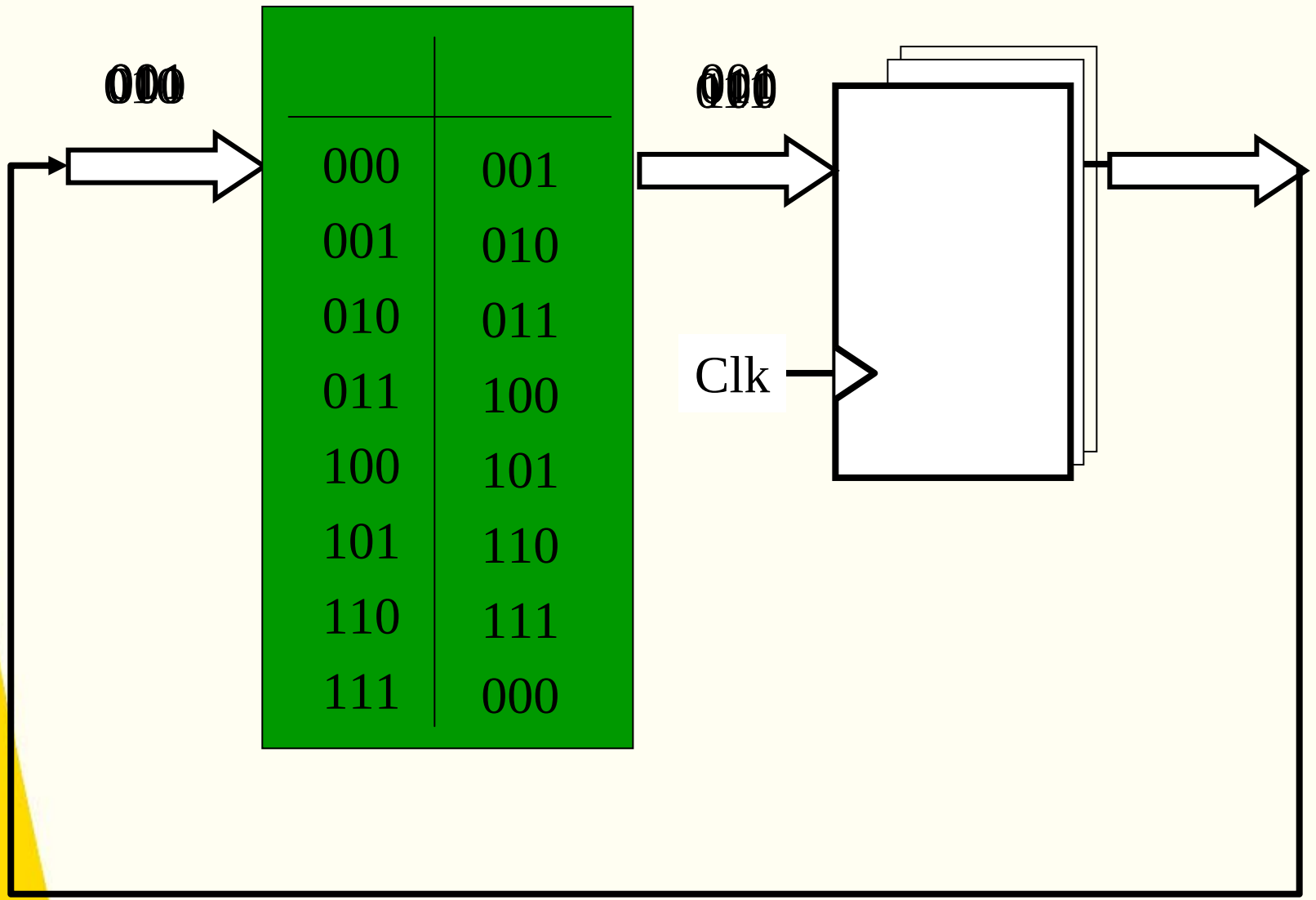
- Stack 8 flip flops to get 8 bit data register
- Strobe Clk to get D3 <= Q;



Register + logic = state machine



A counter

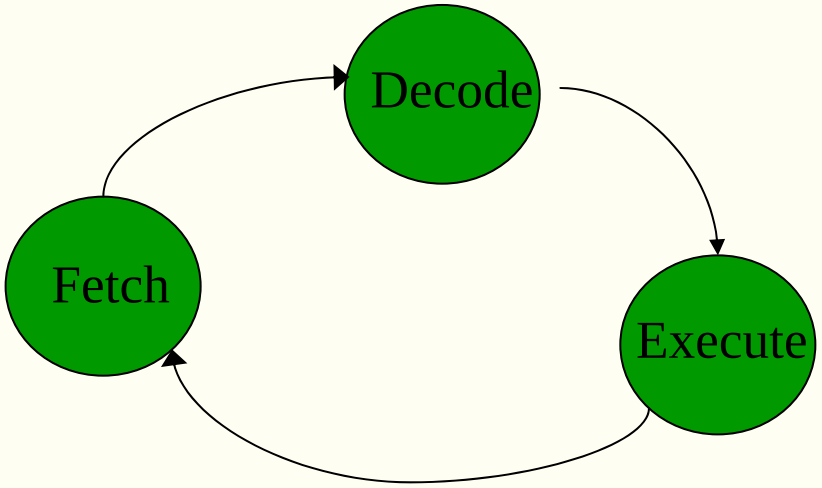

000
001
001
010
010
011
011
100
100
101
101
110
110
111
111
000
Clk

WIMP51 subsystems

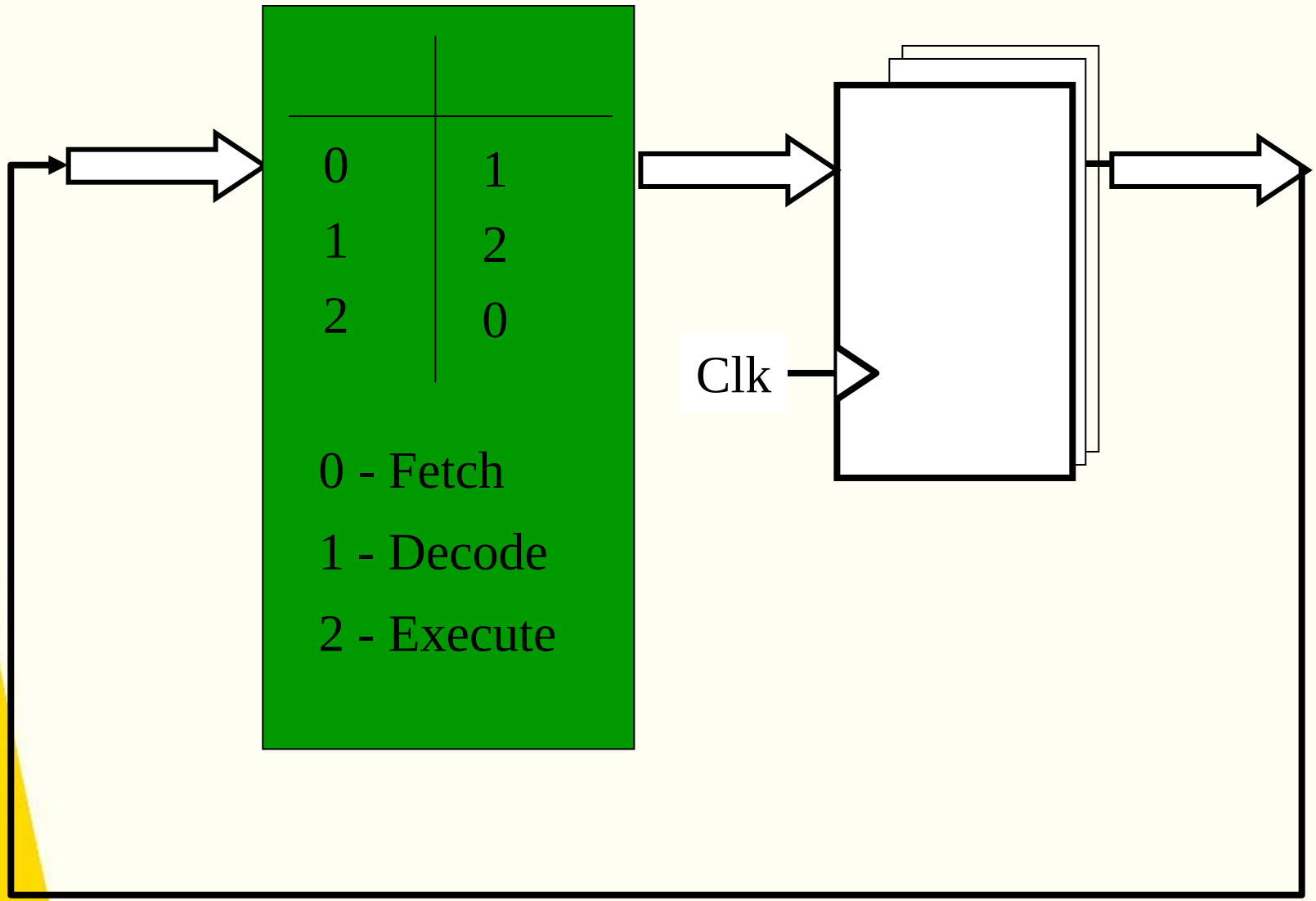
- PC is an eight bit counter, 0 to 255
- AUX register can be added to PC for jump instruction
- ACC and ALU form a ‘register with logic’ block
- IR and memory form a ‘register with logic’ block
- Control Unit (CU) is a 2 bit counter, 0 to 2
- Datapath = IR, AUX, PC, ACC, Register File, ALU, PCALU
  - 12 registers plus Logic (ALU, PCALU)

Wimp51 control unit

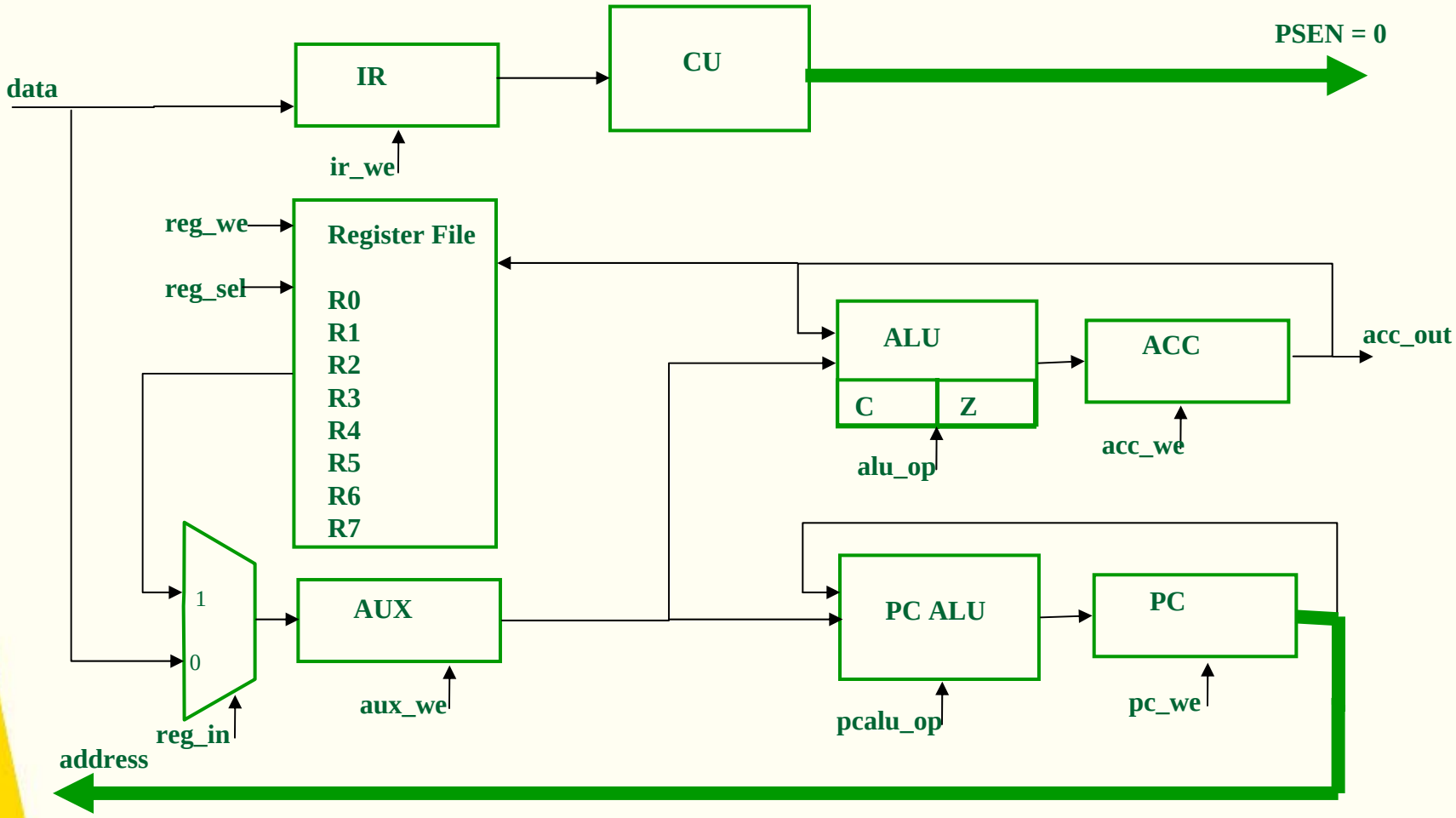
- Simple state machine
- Fetch/Decode/Execute states
- Each state takes one clock cycle
- A complete loop is an instruction cycle



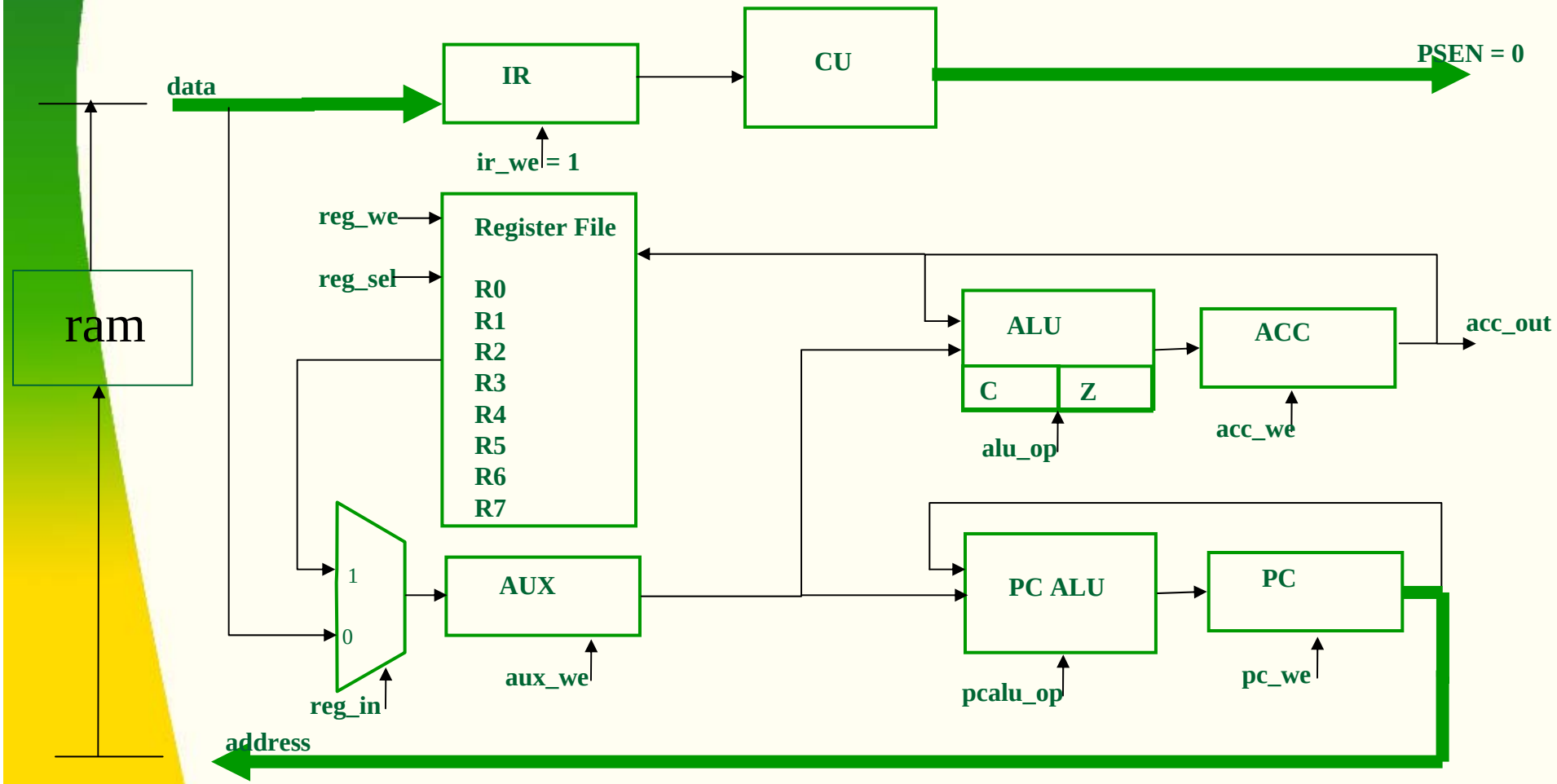
Wimp51 control unit



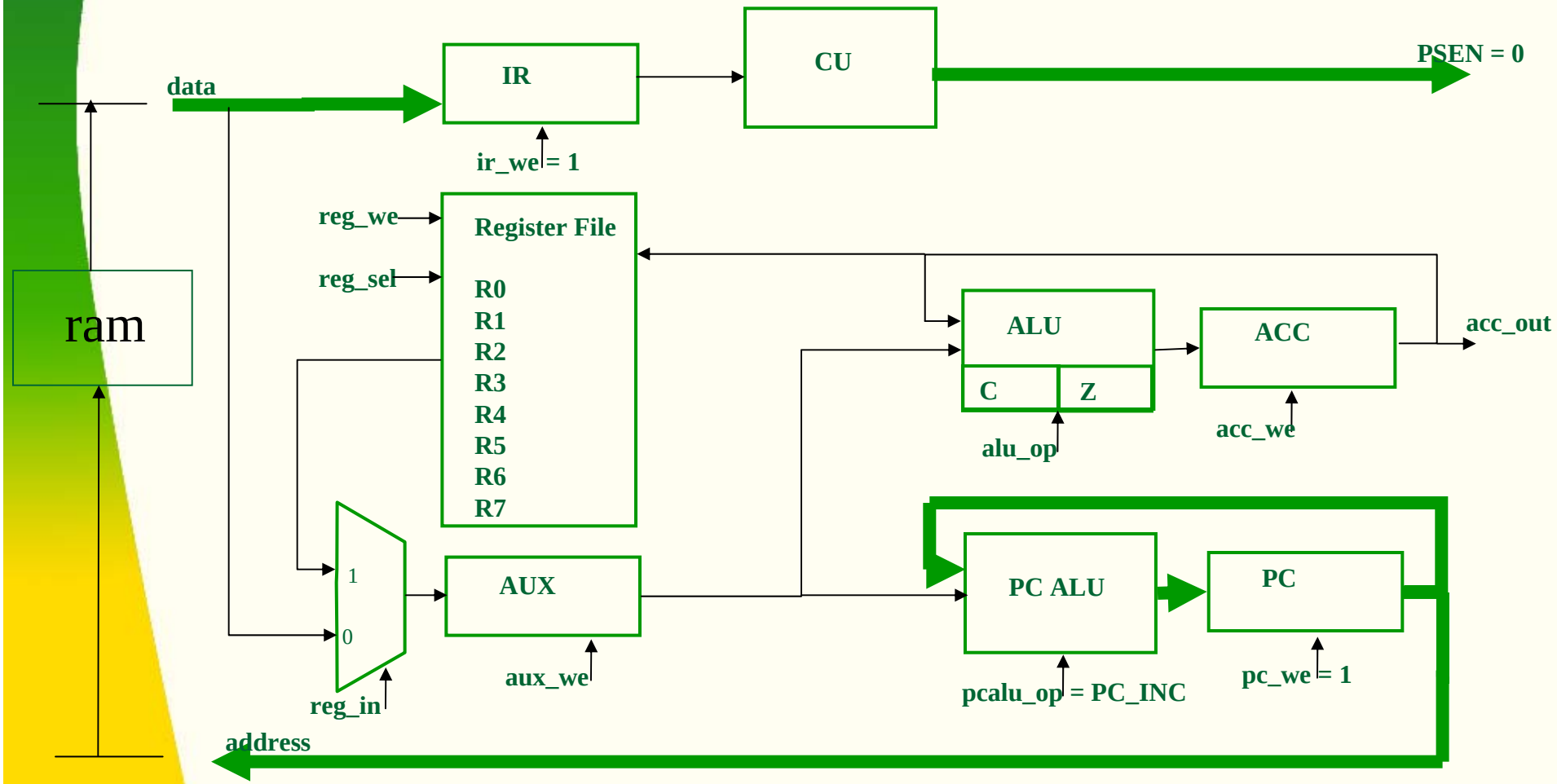
Fetch cycle step 1: addr <= PC



Fetch cycle step 2: IR<= data



Fetch cycle: PC<= PC+1



For Wednesday

- Review today's lecture notes and textbook.
- Download and read WIMP51 handout.
- Begin Assignment 2.