# CISC Design

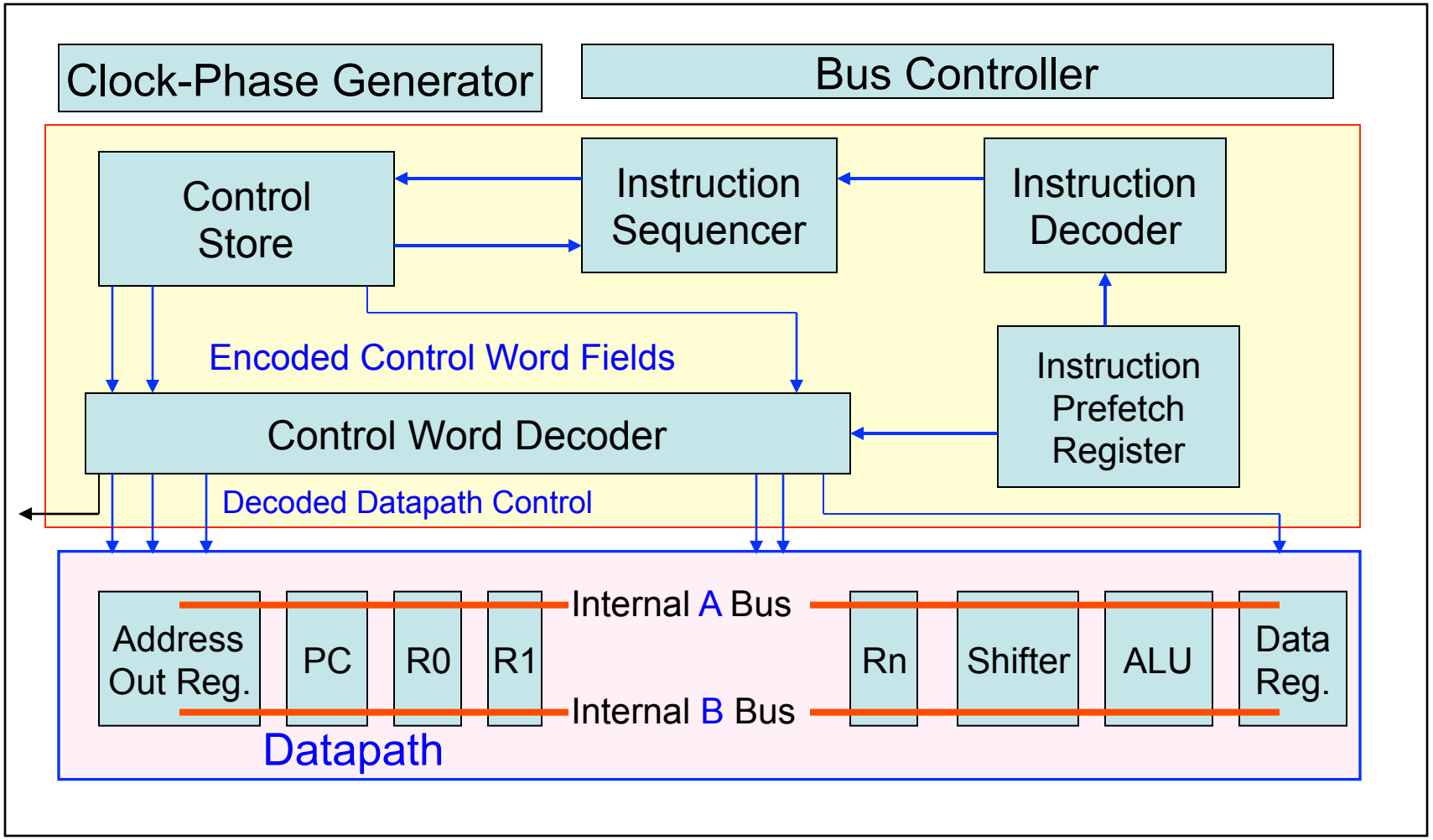
Virendra Singh
Computer Architecture and Dependable Systems Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
http://www.ee.iitb.ac.in/~viren/
E-mail: viren@ee.iitb.ac.in

*EE-309: Microprocessors*

Lecture 21 (03 Sep 2015)

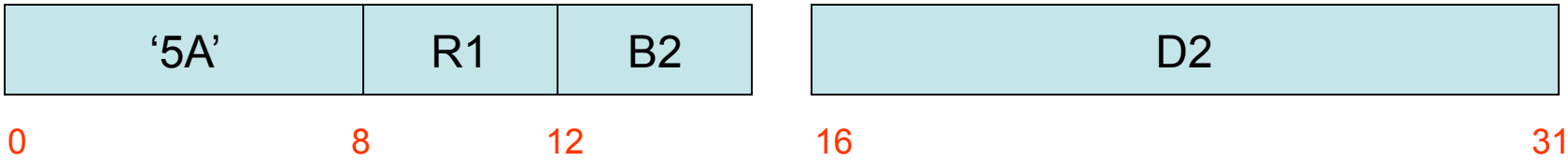
CADSL

# Micro-coded Implementation



# Instruction Execution

ADD R1, D2(B2)




- The second operand is added in the first
- The sum is placed in the first operand location
- The operand and the sum are treated as 16-bit signed binary integers
- The first operand is in the register specified by the R1 field
- The second operand is in the memory – address is calculated by adding the displacement specified by the D2 field to the content of the base register specified by the B2 field

# Execution Steps

## Steps for ADD instruction Execution

1. Fetch the first half word
2. Find ADD control word sequence
3. Fetch the remaining instruction word
4. Calculate the operand address
5. Fetch the operand
6. Add
7. Store the result

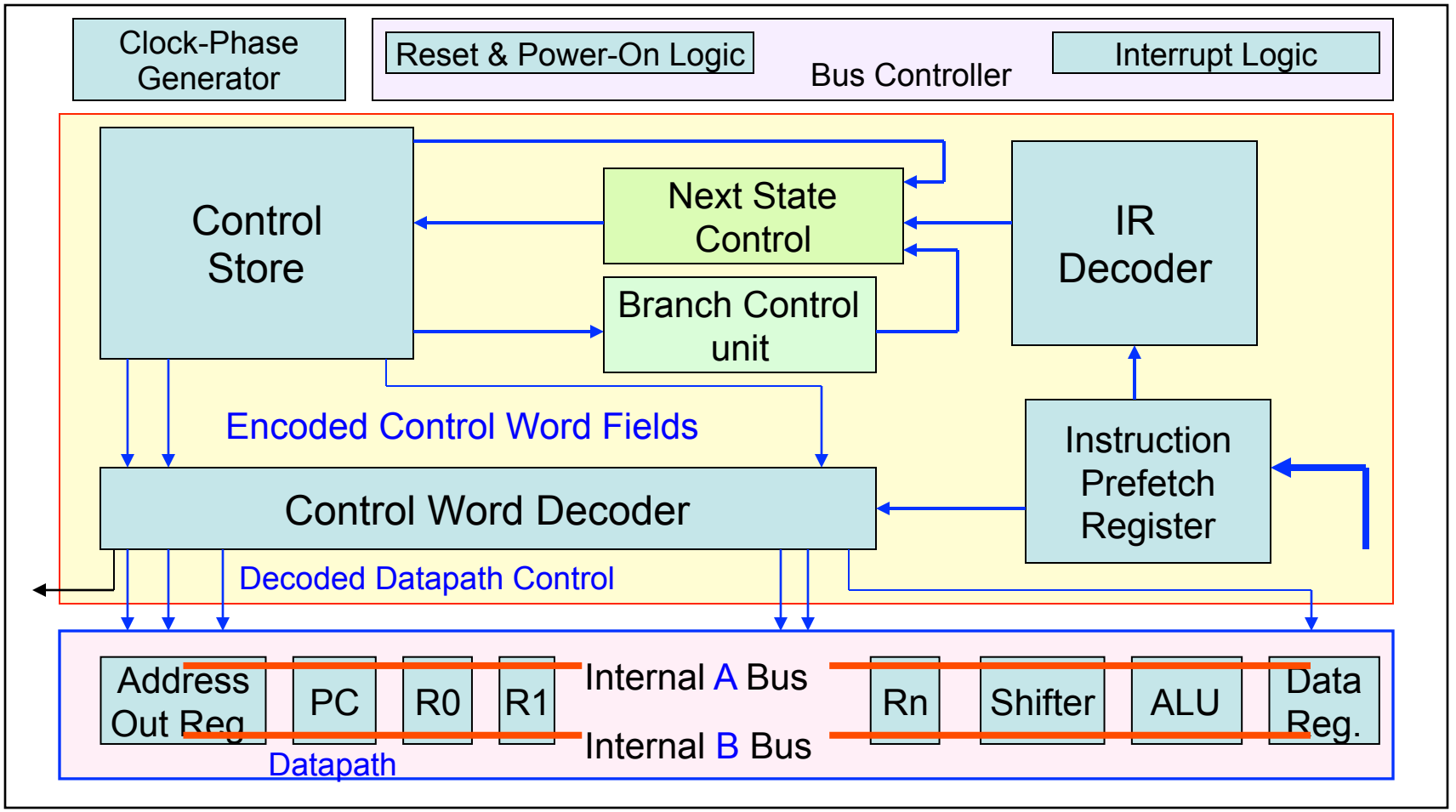
# Execution Steps

1. Fetch the remaining instruction word
   - One state to second half of the ADD instruction
2. Calculate the operand address
   - One state to add D2 displacement and the content of the B2 register
3. Fetch the operand
   - One state to fetch the data half word (put the address on the pads and wait for the operand half-word)
4. Add
   - One state to add the operands
5. Store the result
   - One state to store the result in Register R1

# Execution Steps

6. Update the program counter
   - One state to increment PC
   - One state to save the incremented value
7. Fetch the first half word for the next instruction
   - One state to put the PC value on the pads and wait for the first half of the next instruction
8. Find the address of the next instructions control word sequence
   - One state to put the next instruction into the instruction decoder
9. Branch to the next instruction's control word
   - Zero state – this step is accomplished as a part of the previous step

# Processor - Block Diagram



# Hardware Flowchart

# Hardware Flowchart

## Hardware Flowchart

- Tells us how to get from the architecture to the implementation
- Links programmer's (external) model and the hardware (internal) implementation
- Specify exactly how commands from the instruction set are carried out using Datapath

# Flowchart Objective

- Limit controller size to some fraction of chip area
- Make CPU as fast as possible
- Complete the project as early as possible
- Make the flowcharts easy to translate into hardware

# Thank You