NYU-EL 6463

ADVANCED COMPUTER HARDWARE DESIGN

Processor Design

&

RC5 Implementation

Chi Zhang cz1195

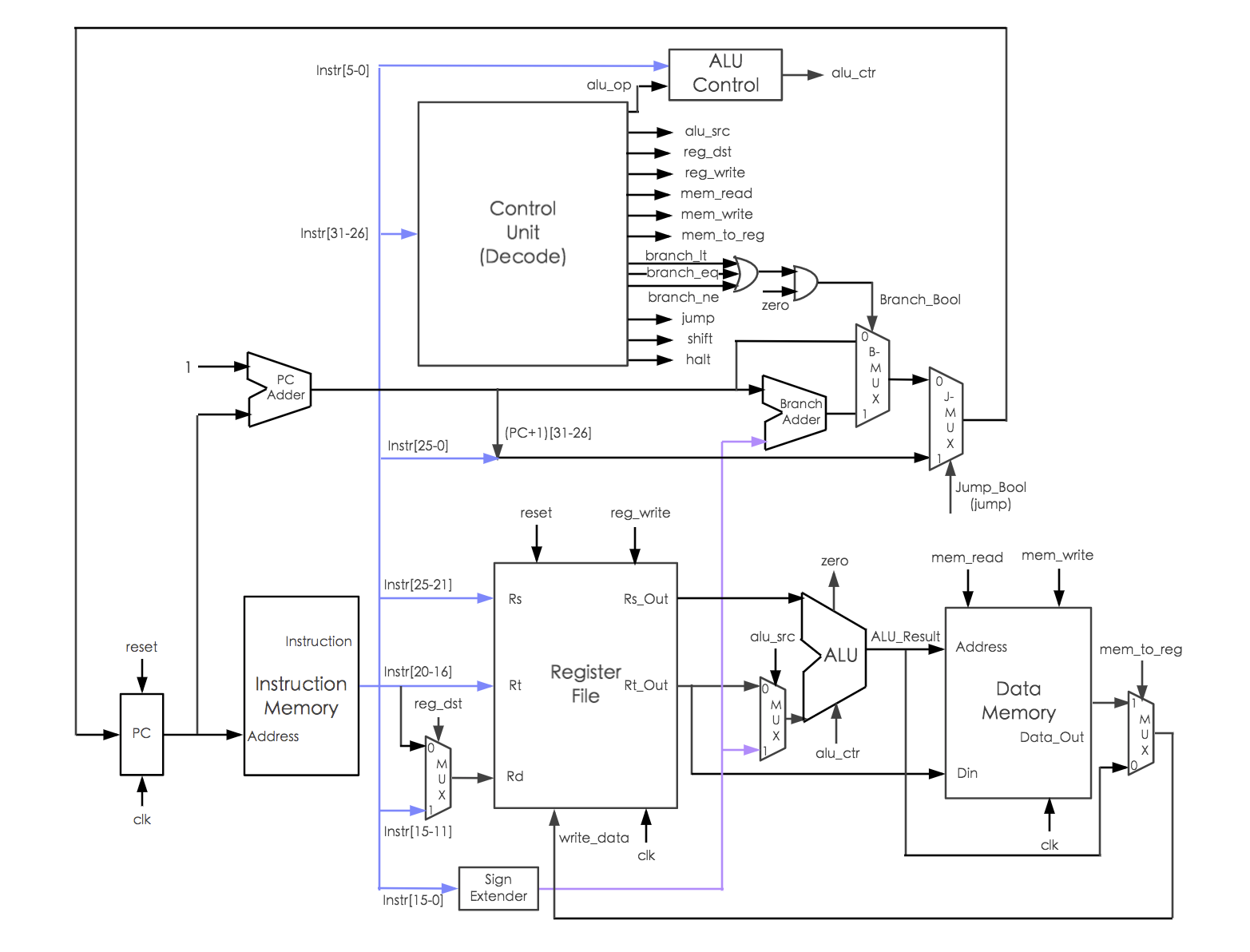
Hao Dong hd864

Jian Song js8612

Wen Da wd580

Zijun Ma zm646

Design block diagram

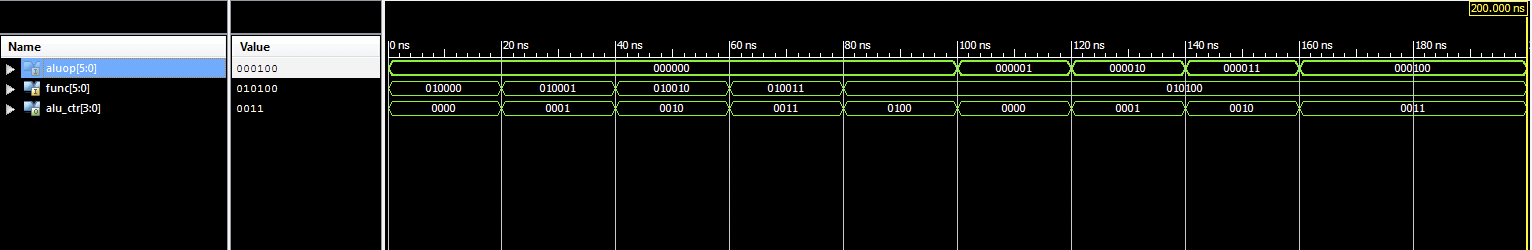


ALU Design

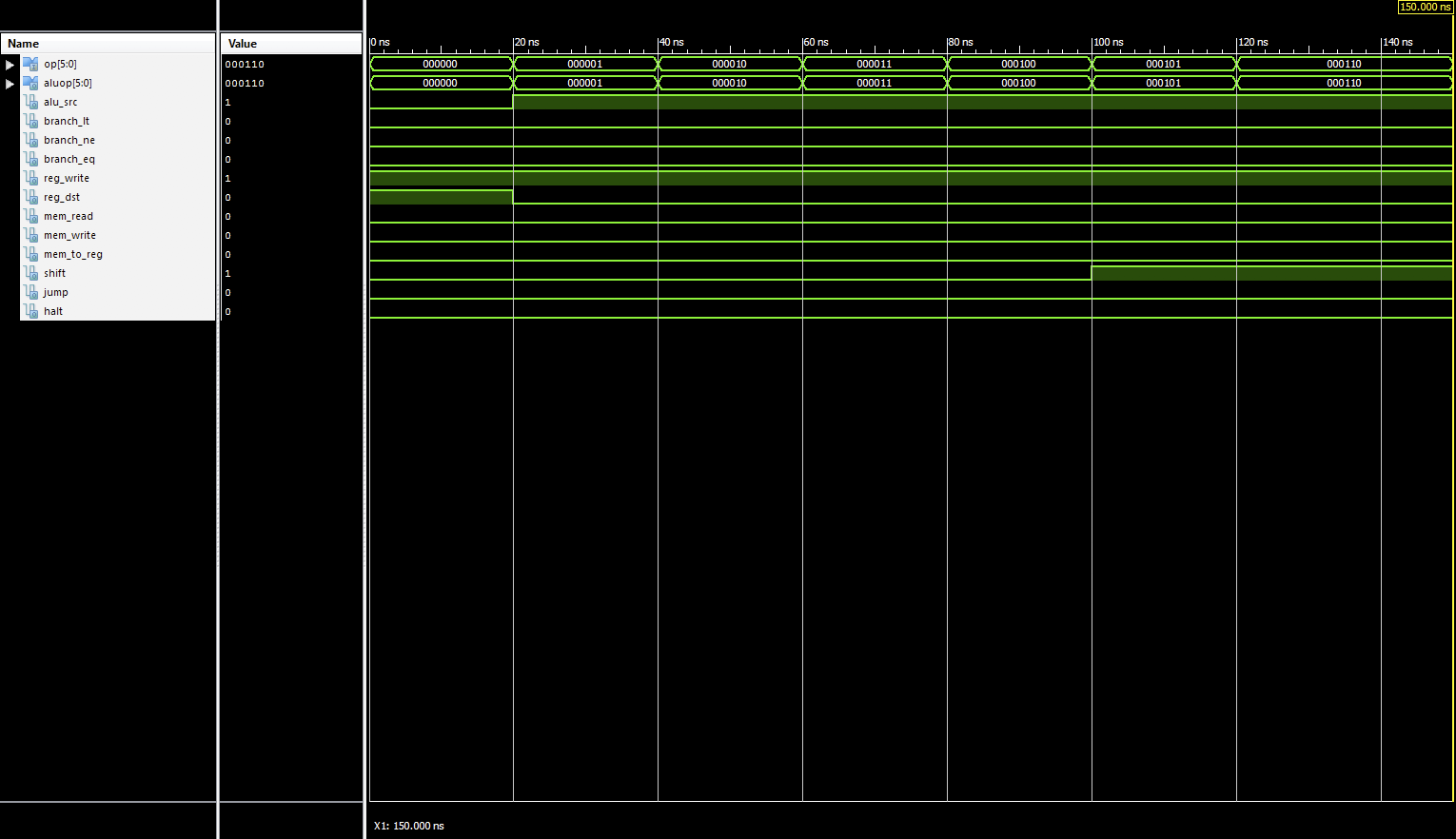


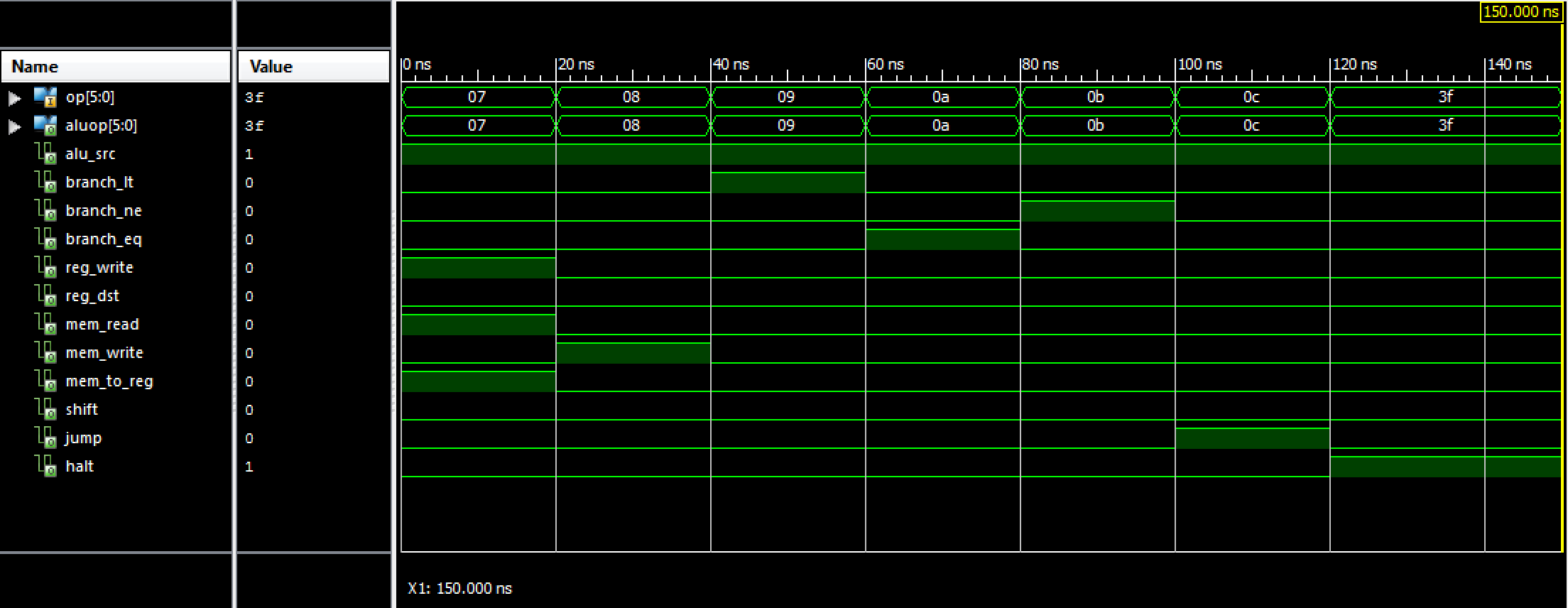
Control Unit Design

ALU Control



Instruction Decode





Complete Processor Design

Register File

Instruction Memory

Data Memory

CPU Verification with Sample Programs

Sample Program 1:

ADDI R1, R0, 7 // R1 = 7

ADDI R2, R0, 8 // R2 = 8

ADD R3, R1, R2 // R3 = R1 + R2 =15

HAL // HALT



Sample Program 2:

This program checks complete instruction set of 18 Instructions.

ADDI R1, R0, 2 // R0=0, R1=2

ADDI R3, R0, 10 // R3=10D (D=Decimal)

ADDI R4, R0, 14 // R4=14D

ADDI R5, R0, 2 // R5=2

SW R4, 2(R3) // 14D is stored in Data memory location 12D

SW R3, 1(R3) // 10D is stored in Data memory location 11D

SUB R4, R4, R3 // R4=2

SUBI R4, R0, 1 // R4=-1D

AND R4, R2, R3 // R4=0

ANDI R4, R2, 10 // R4=0

OR R4, R2, R3 // R4=10D

LW R2, 1(R3) // R2=10D (Loaded back from memory)

ORI R4, R2, 10 // R4=10D

NOR R4, R2, R3 // R4=X“fffffff5”

SHL R4, R2, 10 // R4=X“00002800”

SHR R4, R2, 10 // R4=X“01400000”

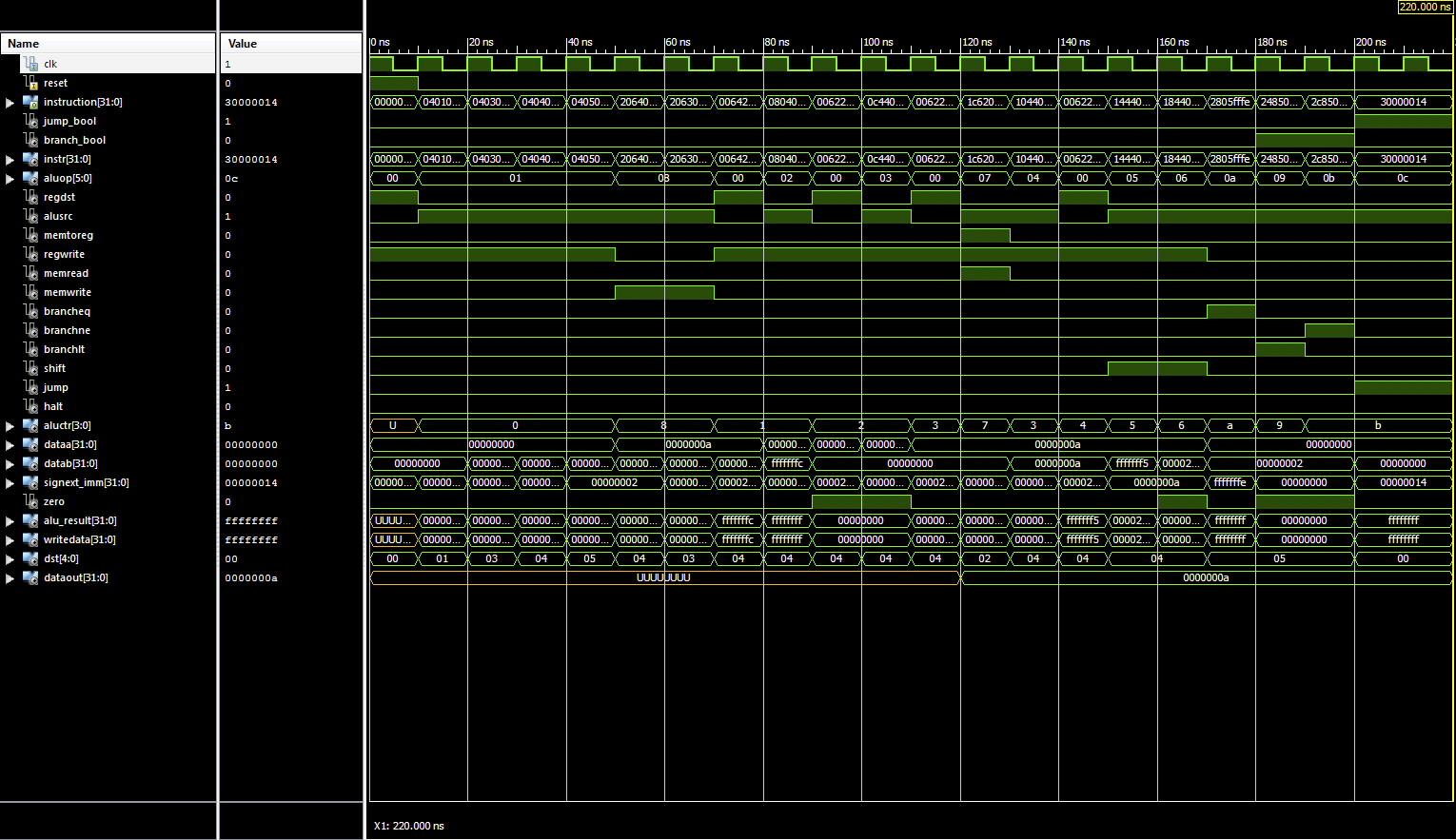
BEQ R5, R0, -2 // No Branch

BLT R5, R4, 0 // Branch to the next instruction

BNE R5, R4, 0 // Branch to the next instruction

JMP 20 // Jump

HAL // Halt



Complete RC5 Assembly code for Encryption and Decryption.

Complete the processor design with additional interfaces and single stepping (7-segment LEDs, Switches for input control).

Complete RC5 Assembly code (Key Expansion, Encryption and Decryption).

High level description of how you implement and run 3 components of RC5 (encryption, decryption and key expansion) on your designed processor

Description of processor interfaces (how you control the inputs and observe the outputs)

Performance and area analysis

Details about how you verified your overall design