Implement a single-cycle MIPS processor for the specifications given below using Verilog.

The Little-Endian Computer Architecture serves as the foundation for the development of the 16-bit NITC-RISC18 computer for teaching. It (NITC-RISC18) is an 8-register, 16-bit computer system. It uses registers R0 to R7 for general purposes. However, Register R7 always stores the program counter. This architecture also uses a condition code register, which has two flags: the Carry flag (C) and the Zero flag (Z).

The NITC-RISC18 is very simple, but it is general enough to solve complex problems with three machine-code instruction formats (R, I, and J type) and a total of six instructions shown below:

R Type Instruction format

Opcode	**Register A (RA)**	**Register B (RB)**	**Register B (RB)**	**Unused**	**Condition (CZ)**
4 bit	3 bit	3 bit	3 bit	1 bit	2 bit

I Type Instruction format

Opcode	**Register A (RA)**	**Register C (RC)**	**Immediate**
4 bit	3 bit	3 bit	6 bits signed

J Type Instruction format

Opcode	**Register A (RA)**	**Immediate**
4 bit	3 bit	9 bits signed

Instructions Encoding

ADD	00_00	RA	RB	RC	0	00
NDU	00_10	RA	RB	RC	0	00
LW	01_00	RA	RB	6 bits immediate		
SW	01_01	RA	RB	6 bits immediate		
BEQ	11_00	RA	RB	6 bits immediate		
JAL	10_00	RA	9 bits immediate offset			

RA: Register A, RB: Register B, RC: Register C

Implement a single-cycle MIPS processor for the above specifications using Verilog.

Mnemonic	Name & Format	Assembly	Action
ADD	ADD (R)	add rc, ra, rb	Add content of regB to regA and store result in regC. It set C and Z flags
NDU	Nand (R)	ndu rc, ra, rb	NAND the content of regB to regA and store result in regC.
LW	Load (I)	lw ra, rb, Imm	Load value from memory into reg A. Memory address is formed by adding immediate 6 bits with content of red B. It sets zero flag.
SW	store (I)	sw ra, rb, Imm	Store value from reg A into memory. Memory address is formed by adding immediate 6 bits with content of red B.
BEQ	Branch on Equality (I)	beq ra, rb, Imm	If content of reg A and regB are the same, branch to PC+Imm, where PC is the address of beq instruction
JAL	Jump and Link (I)	jalr ra, Imm	Branch to the address PC+ Imm. Store PC+1 into regA, where PC is the address of the jal instruction

Naming convention:
The source code and testbench must be named <ROLLNO>_<FIRSTNAME>_<QNUBMER>.v and <ROLLNO>_<FIRSTNAME>_<QNUMBER>_tb.v, respectively.
(Example: BxxyyyyCS_LAXMAN_1.v and BxxyyyyCS_LAXMAN_1_tb.v)