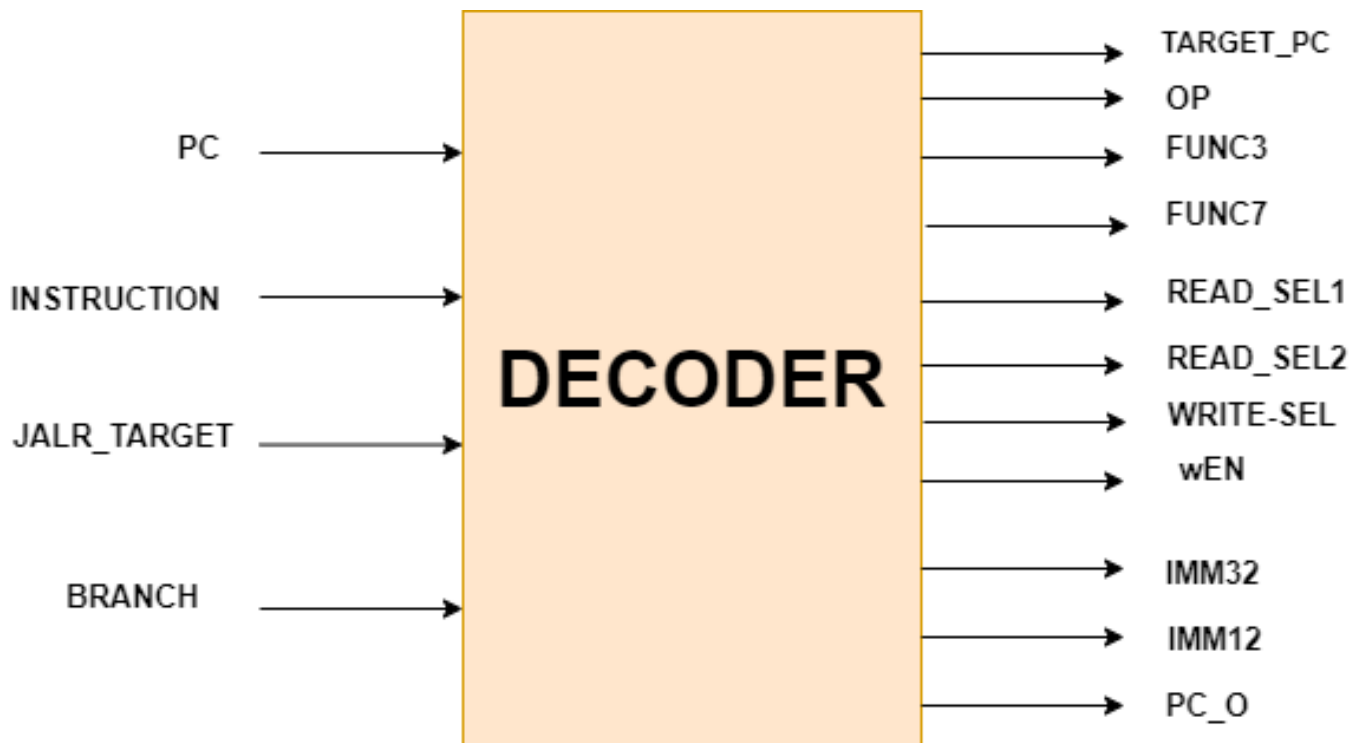# DECODER

Introduction

The decoder in zero-riscy is responsible for decoding the instruction fetched from the instruction memory and generating the control signals.

Problem Statement

- understanding the difference between ISA and their Types

Design and Implementation:

- Block Diagram

- **PC** : signal from fetch module have ADDRESS_BITS { ex : [ADDRESS_BITS-1:0] PC , ADDRESS_BITS = 16 }

- **JALR_TARGET:** signal from ALU , if the instructions JALR ALU send pc which will be fetched

- **BRANCH:** signal from ALU , if the instructions {beq , bne , blt , ...} ALU send signal high or low based on its true or not

- **INSTRUCTION :** The instructions 32 bit { ex : lw instruction 111111111100 01001 010 00110 **0000011** FFC4A303 }

- **TARGET_PC** : signal that have target PC calculations if the instruction is [branch , jal , jalr]   ex: [ADDRESS_BITS-1:0] target pc .

- **OP** : first 7 bits of the instruction and it used to distinguish between difference Types { ex : LOAD 7'b**0000011**}

- **funct3 :** Bits of instruction [14:12]

- **funct7 :** Bits instruction [31:25]

- **READ_SEL1** : operand1 of the instruction { ex : x9 01001 }

- **READ_SEL2 :** operand2 of the instruction { ex : x6 00110 }

- **WRITE-SEL(RD) :** distention register to store the results in GPRs { ex : x5 }

- **wEN :** write Enable for GPRs

- **IMM32** : it's output signal after sign extend {ex : 11111111111111111111111111111100 = -4 }

- **IMM12:** original 12 bit of the imm instruction

- **PC_O** : The same PC input signal

Note : IMM32 Calculated based on the type of the instruction

Table 7.5 *ImmSrc* encoding.

ImmSrc	ImmExt	Type	Description
00	$\{\{20\{Instr[31]\}\}, Instr[31:20]\}$	I	12-bit signed immediate
01	$\{\{20\{Instr[31]\}\}, Instr[31:25], Instr[11:7]\}$	S	12-bit signed immediate
10	$\{\{20\{Instr[31]\}\}, Instr[7], Instr[30:25], Instr[11:8], 1'b0\}$	B	13-bit signed immediate
11	$\{\{12\{Instr[31]\}\}, Instr[19:12], Instr[20], Instr[30:21], 1'b0\}$	J	21-bit signed immediate

31:25	24:20	19:15	14:12	11:7	6:0	
funct7	rs2	rs1	funct3	rd	op	**R-Type**
$imm_{11:0}$		rs1	funct3	rd	op	**I-Type**
$imm_{11:5}$	rs2	rs1	funct3	$imm_{4:0}$	op	**S-Type**
$imm_{12,10:5}$	rs2	rs1	funct3	$imm_{4:1,11}$	op	**B-Type**
$imm_{31:12}$				rd	op	**U-Type**
$imm_{20,10:1,11,19:12}$				rd	op	**J-Type**
fs3	funct2	fs2	fs1	funct3	fd	op	**R4-Type**
5 bits	2 bits	5 bits	5 bits	3 bits	5 bits	7 bits	

Figure B.1 RISC-V 32-bit instruction formats

Interfaces

Signals	Width	interface
PC	12-bit	Fetch module
JALR_TARGET	16-bit	ALU module
BRANCH	1-bit	ALU module
INSTRUCTION	32-bit	Fetch module
TARGET_PC	16-bit	Fetch module
OP	7-bit	Controller
funct3	3-bit	Controller
funct7	7-bit	Controller
READ_SEL1	5-bit	GPR
READ_SEL2	5-bit	GPR , Encryption Accelerator
WRITE-SEL(RD)	5-bit	GPR
wEN	1-bit	GPR
IMM32	32-bit	Pipeline Registers
IMM12	12-bit	Pipeline Registers
PC_O	12-bit	Pipeline Registers