Czech Technical University in Prague, Faculty of Electrical Engineering

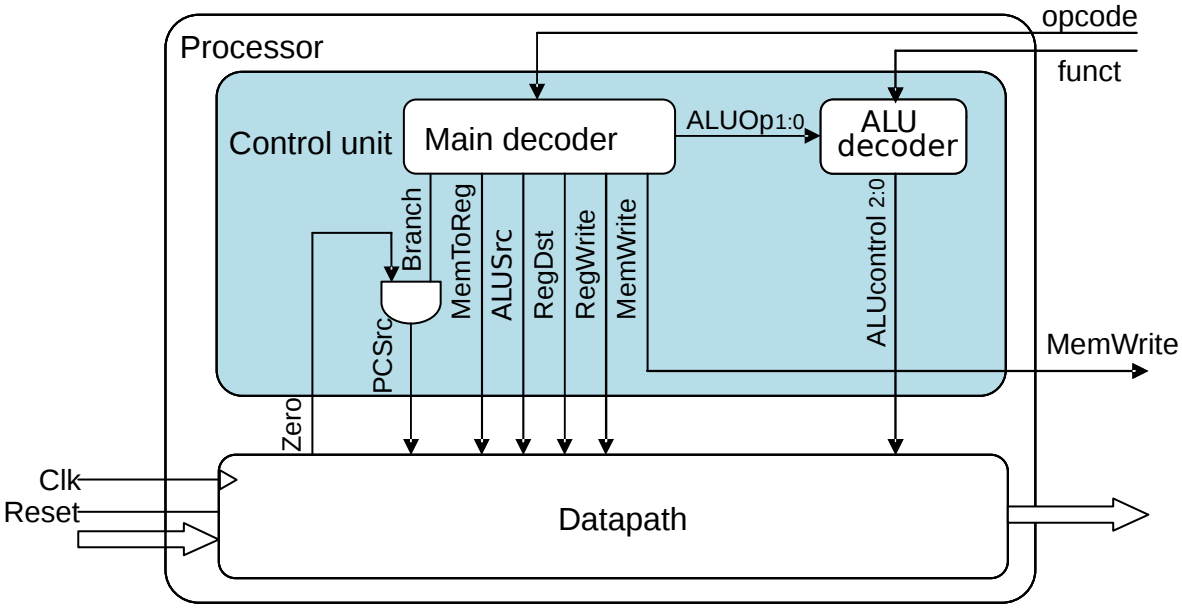
Advanced Computer Architectures (B4M35PAP , BE4M35PAP)

Tasks for Seminar No. 03

1. Logic Circuit Design

The control unit provides reading of the instruction from the memory in the correct order and decoding it, receiving the status messages and ensuring the correct interpretation of the instructions by setting the necessary control signals for the functional units involved in executing the instructions. The operation of the control unit is preferably described by a finite automaton, in particular with multicycle instructions. The individual states of the machine include specific settings of the control signals (the individual phases of the execution of the instruction taking into account the type of operation being performed), the transitions then the conditions (status messages, instruction type, ...) between which the states pass. However, in singlecycle execution of instructions, the operation of the control unit can be expressed using truth tables.

Please implement in Verilog a control unit that meets the specifications below.



Assume that the control unit receives the instruction code - the opcode, the funct field, and only one ALU status message indicating zero of result of ALU operation - zero flag. The control unit is divided into two parts - the main decoder and the ALU decoder. Operation of the control unit is described in the following tables. The main decoder and ALU decoder are connected by ALUOp.

Tab. 1 Main decoder

Input	Output						
Opcode	RegWrite	RegDst	ALUSrc	ALUOp	Branch	MemWrite	MemToReg
000000	1	1	0	10	0	0	0
100011	1	0	1	00	0	0	1
101011	0	0	1	00	0	1	0
000100	0	0	0	01	1	0	0
001000	1	0	1	00	0	0	0

Tab.2 ALU decoder

Input		Output
ALUOp	Funct	ALUControl
00	X	010
01	X	110
1X	100000	010
1X	100010	110
1X	100100	000
1X	100101	001
1X	101010	111

Furthermore, let PCSrc control signal (PCSrc be the output signal of the controller) is set if the signal of the main decoder Branch signal is set as well as the Zero status signal. The schema of the control unit, along with a representation of the interaction with the data path of the processor (datapath) is shown in the opening image.

2. Program Design

Suppose you have instructions add, sub, and, or, slt, add, lw, sw, beq. Rewrite the program below in HLL using these instructions.

```
if (a <= b)
    for (int i = 0; i! = c; i ++)
        a + = i;
else
    a = a-b;
while (1);
```

Assume that variables a, b and c are int types (4B) and are already stored in memory at 0x0010, 0x0014 and 0x0018.
Note: After successful execution of the program, the variable will be modified accordingly and at 0x0010 a the program hangs in an infinite loop.

Add	add $d,$s,$t	$d = $s + $t	Arithmetický sum
Subtract	sub $d,$s,$t	$d = $s - $t	Arithmetic subtraction
Add immediate	addi $t,$s,C	$t = $s + C (signed)	Addition of sign extended constant
And	and $d,$s,$t	$d = $s & $t	Logic and
Or	or $d,$s,$t	$d = $s | $t	Logic or
Set on less than	slt $d,$s,$t	$d = ($s < $t)	Test condition if the first register is smaller than the second; $d is set to 0 or 1
Load word	lw $t,C($s)	$t = Memory[$s + C]	Loads word (4 B) from memory to register. MEM[$s+C] and next 3 bytes
Store word	sw $t,C($s)	Memory[$s + C] = $t	Stores word (4 B) in register to memory. MEM[$s+C] and nex 3 bytes
Branch on equal	beq $s,$t,C	if ($s == $t) go to PC+4+4*C	Branches/sets PC to relative address if values in imput registers are equal

The register letters d, t, and s are placeholders for (register) numbers or register names. C denotes a constant (immediate). register letters d, t and s are placeholders for (register) numbers or register names.