

Zadanie 1

1.

```
000000 SSSSS TTTT DDDDD <<<< FFFFFF
000000 SSSSS 00000 00000 00000 001000
```

2.

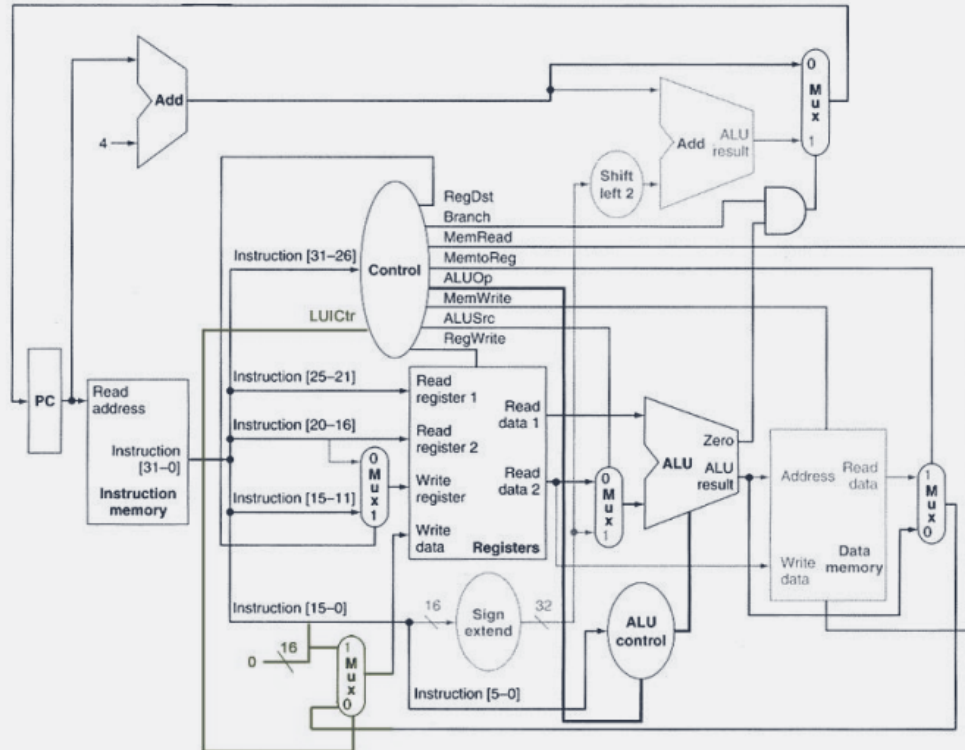
```
000000 SSSSS TTTT CCCCCCCCCCCCCC
001111 00000 TTTT CCCCCCCCCCCCCC
```

3.

Dodatkowe ALU do dodania 4 oraz dodatkowe piny w Register File, Write2 i WriteData2. Ustalamy, że rejestr do którego wpisujemy wartość jest taki sam jak ReadRegister1 (A1)

Inne

5.10 Show the needed changes to the above single cycle processor design of MIPS shown below to support the load upper immediate LUI instruction of the MIPS instruction set architecture.



Instr	RegDst	ALUSrc	Mem toReg	Reg Write	Mem Read	Mem Write	Branch	ALUOp 1	ALUOp 2	LUICtr
R-type	1	0	0	1	0	0	0	1	0	0
lw	0	1	1	1	1	0	0	0	0	0
sw	x	1	x	0	0	1	0	0	0	0
beq	x	0	x	0	0	0	1	0	1	0
lui	0	x	x	1	x	0	0	x	x	1
LUI (ALT)	0	1	0	1	x	0	0	1	1	


Alternatively the other solution (LUI ALT) would not require any changes to the data path and would feed the zero extended (Extend=0) of Instruction[15-0] to the second input of the ALU (ALUSrc=1), set the ALUOp bits=11 for a logical 16-bit left shift operation at the ALU specified by the ALU control, then set MemtoReg=0, set RegWrite=1 and set RegDst=0 to write the ALU result to Reg Rt.

Figure 1:

- **ALU control input**

000	AND
001	OR
010	add
110	subtract
111	set-on-less-than

Figure 2:


Usuwanie hazardu R-A-W (2)

- Wstrzymywanie potoku po wykryciu hazardu
 - układ kombinacyjny porównuje numery rejestrów źródłowych instrukcji w stopniu RD z numerami rejestrów docelowych instrukcji w stopniu ALU i MEM
 - w przypadku wykrycia zgodności, instrukcja zostaje zatrzymana w stopniu RD
 - stopnie IF i RD „stoja”
 - pozostałe stopnie pracują normalnie, ze stopnia RD do ALU jest „wstrzykiwana” instrukcja pusta
 - program wykonuje się poprawnie, bez konieczności dodawania instrukcji pustych
 - zależności między instrukcjami powodują opóźnienia
 - w programach często występują sekwencje instrukcji zależnych

Figure 3: