

Procesor MIPS

1) Instructiuni alese

Tip R:

Xor : $RF[rd] \leftarrow RF[rs] \wedge RF[rt]$
 xor \$5, \$1, \$3
 b"000000_00001_00011_00101_00000_000111"

Nand: $RF[rd] \leftarrow RF[rs] \text{ nand } RF[rt]$
 nand \$5, \$1, \$3
 b"000000_00001_00011_00101_00000_000001"

Tip I:

Andi: $RF[rt] \leftarrow RF[rs] \& Z_Ext(imm)$
 andi \$1, \$2,
 b"000011_00010_00001_00000000000000101"

Ori: $RF[rt] \leftarrow RF[rs] | Z_Ext(imm)$
 ori \$1, \$2, 3
 b"000101_00010_00001_00000000000000011"

2) Tabel semnale de control

Instr	Opcode	RegDst	ExtOp	AluSrc	Branch	Jump	MemWrite	MemtoReg	RegWrite	AluOp	func	ALUctrl
ADD	000000	1	X	0	0	0	0	0	1	000	000000	000(+)
SUB	000000	1	X	0	0	0	0	0	1	000	000010	010(-)
SLL	000000	1	X	0	0	0	0	0	1	000	000011	011(<<)
SRL	000000	1	X	0	0	0	0	0	1	000	000100	100(>>)
AND	000000	1	X	0	0	0	0	0	1	000	000101	101(&)
OR	000000	1	X	0	0	0	0	0	1	000	000110	110()
XOR	000000	1	X	0	0	0	0	0	1	000	000111	111(^)
NAND	000000	1	X	0	0	0	0	0	1	000	000001	001(NAND)
ADDI	000001	0	1	1	0	0	0	0	1	001(+)	-	000(+)
ORI	000010	0	0	1	0	0	0	0	1	010()	-	110()
ANN DI	000011	0	0	1	0	0	0	0	1	011(&)	-	101
LW	000100	0	1	1	0	0	0	1	1	100	-	000(+)
SW	000101	X	1	1	0	0	1	X	0	101	-	000(+)
BEQ	000110	X	1	0	1	0	0	0	0	110	-	010(-)
JUMP	000111	X	X	X	X	1	0	X	0	111	-	XXX

3) Program implementat

#4010001: addi \$1, \$0, 1	# Inițializează contorul buclei (\$1) la 1
#4020007: addi \$2, \$0, 7	# Inițializează indexul memoriei (\$2) la 7
#4030000: addi \$3, \$0, 0	# Încarcă valoarea N în \$3
#4040000: addi \$4, \$0, 0	# Inițializează suma la 0
#C250001: lw \$5, 1(\$1)	# Încarcă elementul curent din memorie în \$5
#C050002: lw \$5, 2(\$0)	# Încarcă elementul curent din memorie în \$5
#812000: lw \$4, 0(\$1)	# Încarcă elementul curent din memorie în \$4
#1C00000B: beq \$0, \$0, 11	# Sare la Sfârșit dacă condiția este îndeplinită
#222800: sw \$5, 0(\$2)	# Stochează rezultatul curent în memorie
#20800: sw \$0, 0(\$0)	# Stochează 0 în locația de memorie 0
#872000: sw \$7, 0(\$4)	# Stochează rezultatul în memorie
#4210001: addi \$1, \$1, 1	# Incrementare contor buclă
#4630001: addi \$3, \$3, 1	# Incrementare index memorie
#18410001: lw \$4, 1(\$2)	# Încarcă elementul curent din memorie în \$4
#1C000004: beq \$0, \$0, 4	# Sare la începutul buclei