INSTRUCTION	OPCODE	TYPE	DESCRIPTION	OPERATION	INSTRUCTION CODING (21-bit BINARY)					VALUE OF THE PC AFTER INSTRUCTION IS EXECUTED	Control Unit signal values based on OpCode							
											RegSrc(2)	ALUOp(3)	RegWrite(1)	Write7Seg(1)	WriteLEDs(1)	PCInc(1)	Beq(1)	JiJr(2)
add rs rt rd	0000	Arith	Add the contents of the two registers denoted by the rs and rt fields and store the result in the register indicated in the rd field	rd = rs + rt	0000	rs	rt	rd	xxxxxxxx	PC = PC + 1	10	000	1	0	0	0	0	00
sub rs rt rd	0001	Arith	Subtract the contents of the register denoted by the rt field from the register denoted by the rs field and store the result in the register indicated in the rd field	rd = rs - rt	0001	rs	rt	rd	xxxxxxxx	PC = PC + 1	10	001	1	0	0	0	0	00
and rs rt rd	0010	Logic	And the contents of the two registers denoted by the rs and rt fields and store the result in the register indicated in the rd field	rd = rs AND rt	0010	rs	rt	rd	xxxxxxxx	PC = PC + 1	10	010	1	0	0	0	0	00
or rs rt rd	0011	Logic	Or the contents of the two registers denoted by the rs and rt fields and store the result in the register indicated in the rd field	rd = rs OR rt	0011	rs	rt	rd	xxxxxxxx	PC = PC + 1	10	011	1	0	0	0	0	00
xor rs rt rd	0100	Logic	Xor the contents of the two registers denoted by the rs and rt fields and store the result in the register indicated in the rd field	rd = rs XOR rt	0100	rs	rt	rd	xxxxxxxx	PC = PC + 1	10	100	1	0	0	0	0	00
not rs rd	0101	Logic	Invert the contents of the register denoted by the rs field and store the result in the register indicated in the rd field	rd = NOT rs	0101	rs	xxx	rd	xxxxxxxx	PC = PC + 1	10	101	1	0	0	0	0	00
shl rs rd	0110	Arith	Shift the contents of the register denoted by the rs field left one position (filling the vacated slot on the right with a zero) and store the result in the register indicated in the rd field	rd = ⟵ rs	0110	rs	xxx	rd	xxxxxxxx	PC = PC + 1	10	110	1	0	0	0	0	00
shr rs rd	0111	Arith	Shift the contents of the register denoted by the rs field right one position (filling the vacated slot on the left with a zero) and store the result in the register indicated in the rd field	rd = ⟶ rs	0111	rs	xxx	rd	xxxxxxxx	PC = PC + 1	10	111	1	0	0	0	0	00
ld rd i	1000	Other	Load the register denoted by the rd field with the contents of the immediate field of the instruction	rd = LOAD i	1000	xxx	xxx	rd	iiiiiiii	PC = PC + 1	00	xxx	1	0	0	0	0	00
spc rd	1001	Other	Store the current contents of the 8-bit program counter in the register indicated in the rd field	rd = PC	1001	xxx	xxx	rd	xxxxxxxx	PC = PC + 1	01	xxx	1	0	0	0	0	00
beq rs rt i	1010	Other	If the contents of the two registers denoted by the rs and rt fields are equal then increment the program counter with the value in the immediate field	rs - rt	1010	rs	rt	xxx	iiiiiiii	if(Zero = 1) then PC+i else PC = PC+1	xx	001	0	0	0	0	1	00
ji i	1011	Other	Reset the program counter to the value found in the immediate field	PC = i	1011	xxx	xxx	xxx	iiiiiiii	PC = i	xx	xxx	0	0	0	0	0	10
jr rs	1100	Other	Reset the program counter to the value stored in the register denoted by the rs field	PC = rs	1100	rs	xxx	xxx	xxxxxxxx	PC = rs	xx	xxx	0	0	0	0	0	01
w7seg rs	1101	Other	Write the contents of the register denoted by the rs field into another 8-bit register connected to the seven segment displays	SegH + Seg L = rs	1101	rs	xxx	xxx	xxxxxxxx	PC = PC + 1	xx	xxx	0	1	0	0	0	00
wleds rs	1110	Other	Write the contents of the register denoted by the rs field into another 8-bit register connected to the 8 LEDs	LEDs = rs	1110	rs	xxx	xxx	xxxxxxxx	PC = PC + 1	xx	xxx	0	0	1	0	0	00
rsw rd	1111	Other	Read the value of the least significant 8-switches and store the contents in the register denoted by the rd field	rd = switches(7 : 0)	1111	xxx	xxx	rd	iiiiiiii	PC = PC + 1	11	xxx	1	0	0	0	0	00