

Documentation Technique pour le Jeu d'Instruction 8 Bits

Introduction

Ce jeu d'instruction 8 bits est conçu pour la Machine virtuelle LogiOs simulée dans logisim

Structure des Instructions

Modèle d'Instruction de 4 Octets

Bit (24-31)	Bit (16-23)	Bit (8-15)	Bit (0-7)
Pilotage	Registre A	Registre B	Output

1er octect poids fort Bit (24-31)

Bit (30-31)		Bit (28-29)		Bit (24-27)				
Fonction		<table><tr><td>0: Valeur direct</td><td>1: Adresse de registre</td></tr><tr><td colspan="2">Pilote la valeur attendue dans les registres A et B</td></tr></table>		0: Valeur direct	1: Adresse de registre	Pilote la valeur attendue dans les registres A et B		Opcode
0: Valeur direct	1: Adresse de registre							
Pilote la valeur attendue dans les registres A et B								
ALU	00							
MOV	01							
COMPARE	10							
JMP	11							

Opcode Bit (24-27)

Méthode	Code	commentaire	
ALU			
OR	0000	Écrit le resultat dans le registre à l'adresse lue dans Output Bit(0-7)	
AND	0001		
XOR	0010		
NOR	1000		
NAND	1001		
NXOR	1010		
ADD	0011		
SUB	0100		
MOV			
MOV	0000	...	
COMPARE			
GT	0000	>	Écrit le resultat dans le registre R1
EQ	0001	=	
LT	0010	<	
NGT	0100	<=	
NEQ	0101	!=	
NLT	0110	>=	

JMP			
JGZ	0000	R1 > 0	Saute à l'instruction à l'adresse lue dans le Registre A
JEZ	0001	R1 = 0	
JLZ	0010	R1 < 0	
JNGZ	0100	R1 <= 0	
JNEZ	0101	R1 != 0	
JNLZ	0110	R1 >= 0	

Détails Techniques

Registres

Nom	Adresse
R1	0x00
R2	0x01
R3	0x02
R4	0x03
R5	0x04
R6	0x05
R7	0x06

Exemples de Programmes

Multiplier 6 x 8, et écrire le résultat dans R3

```
mov 6 r1
mov 8 r2
mov 0 r3
:loop
    sub r1 1 r1
    add r2 r3 r3
    jnez loop
    jmp endloop
:end
    stop
:endloop
    jmp end
```

LABEL	
:<NOM>	