Projet PARM

Polytech ARM-based embedded processor

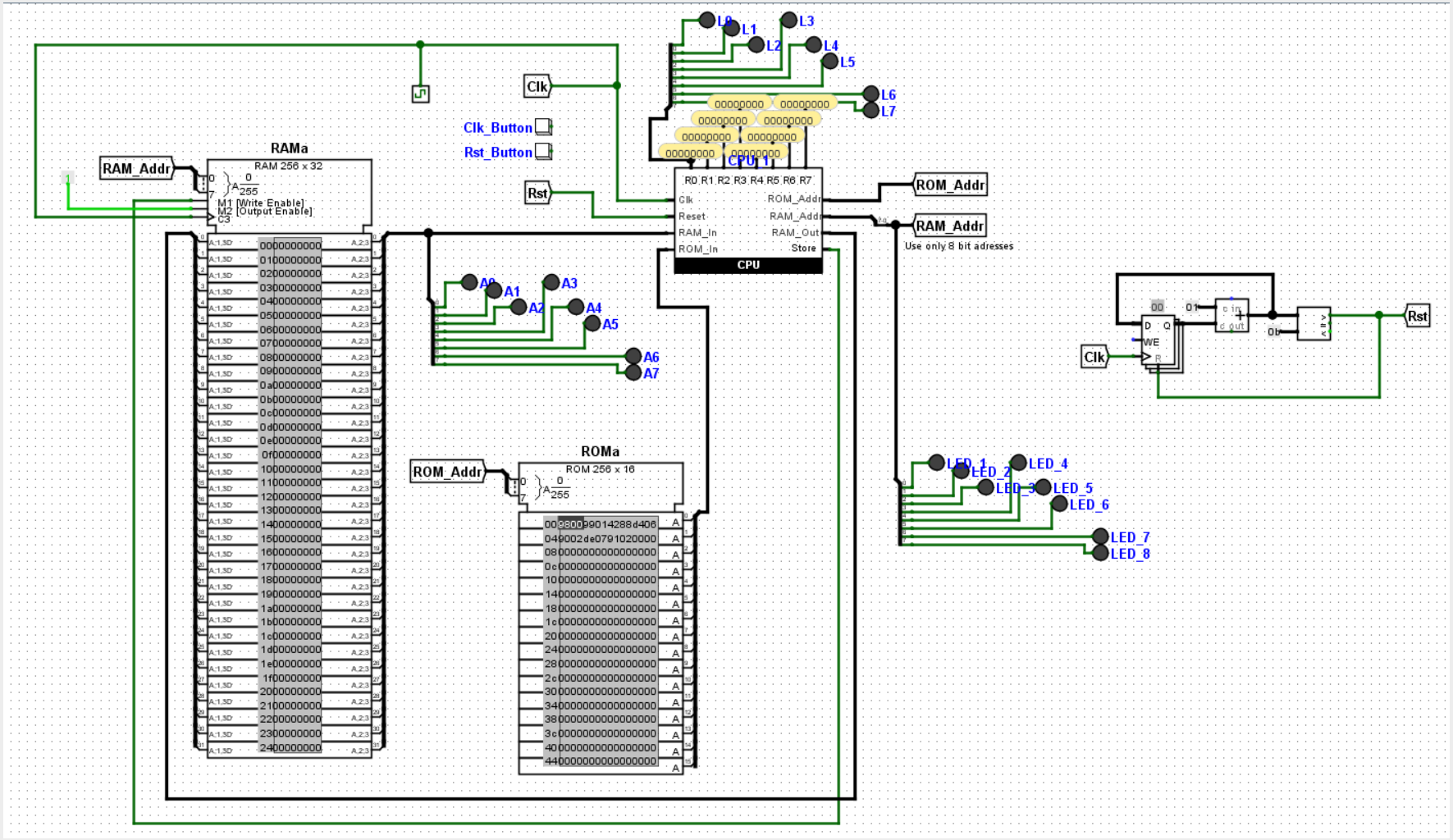
By Arms-and-Legs

Sommaire

1

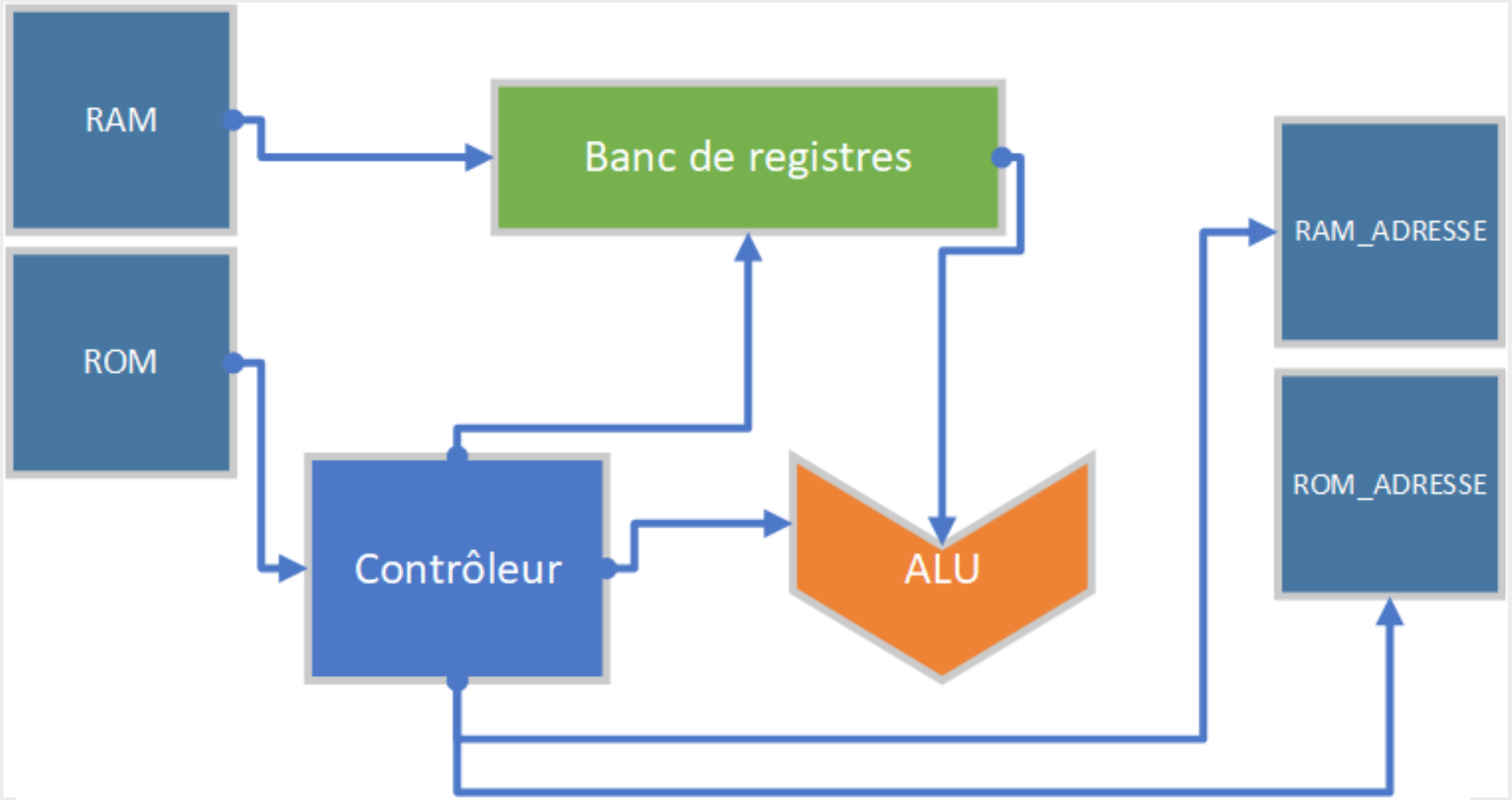
Les Grandes Lignes

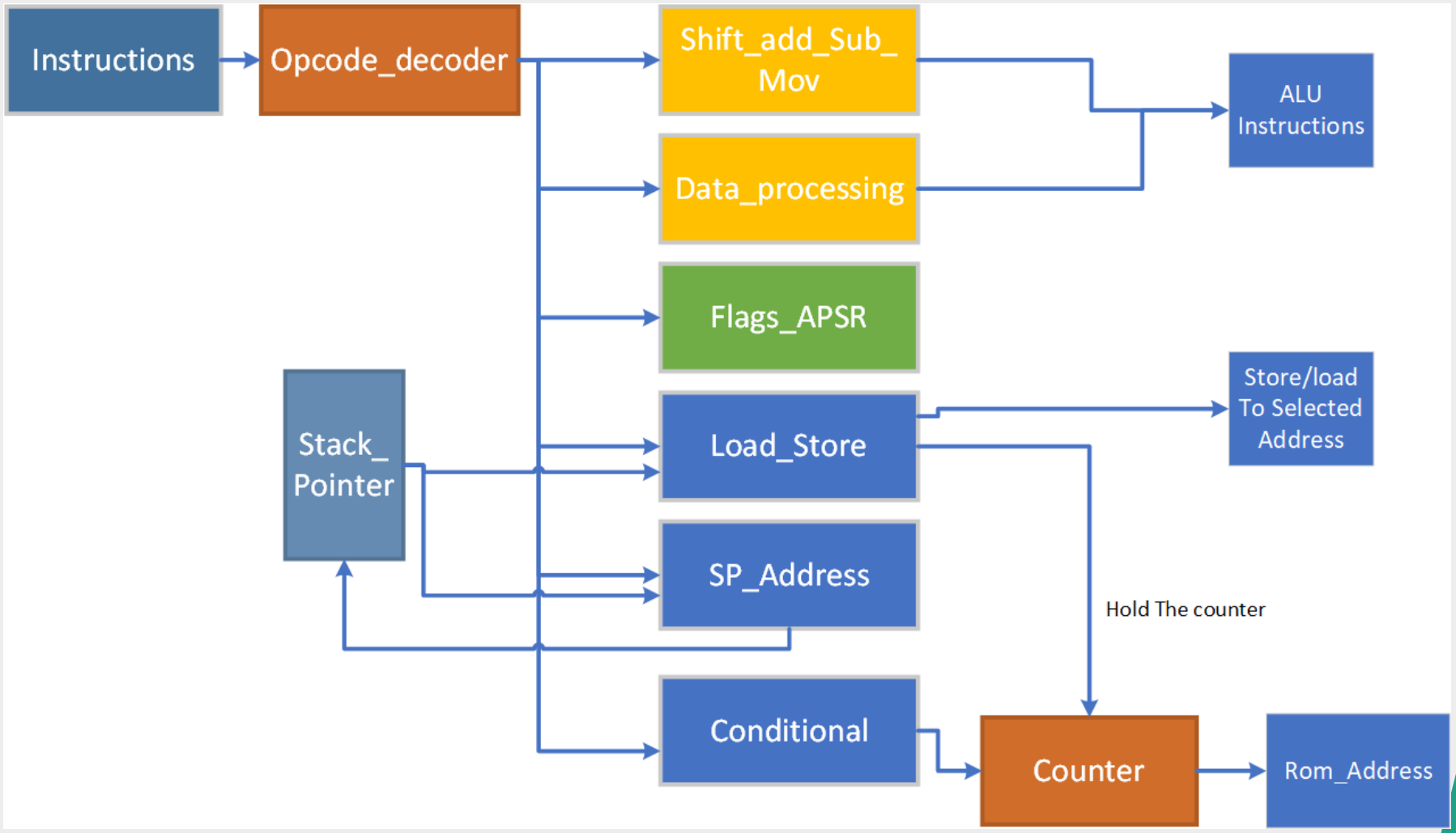
Machine


Clk
Clk_Button
Rst_Button
Rst
RAMa
RAM 256 x 32
RAM_Addr
M1 [Write Enable]
M2 [Output Enable]
CPU 1
R0 R1 R2 R3 R4 R5 R6 R7
Clk
Reset
RAM_In
ROM_In
ROM_Addr
RAM_Addr
RAM_Out
Store
CPU
ROM_Addr
RAM_Addr
Use only 8 bit adresses
ROMa
ROM 256 x 16
ROM_Addr
L0
L1
L2
L3
L4
L5
L6
L7
A0
A1
A2
A3
A4
A5
A6
A7
LED_1
LED_2
LED_3
LED_4
LED_5
LED_6
LED_7
LED_8
Clk
Rst

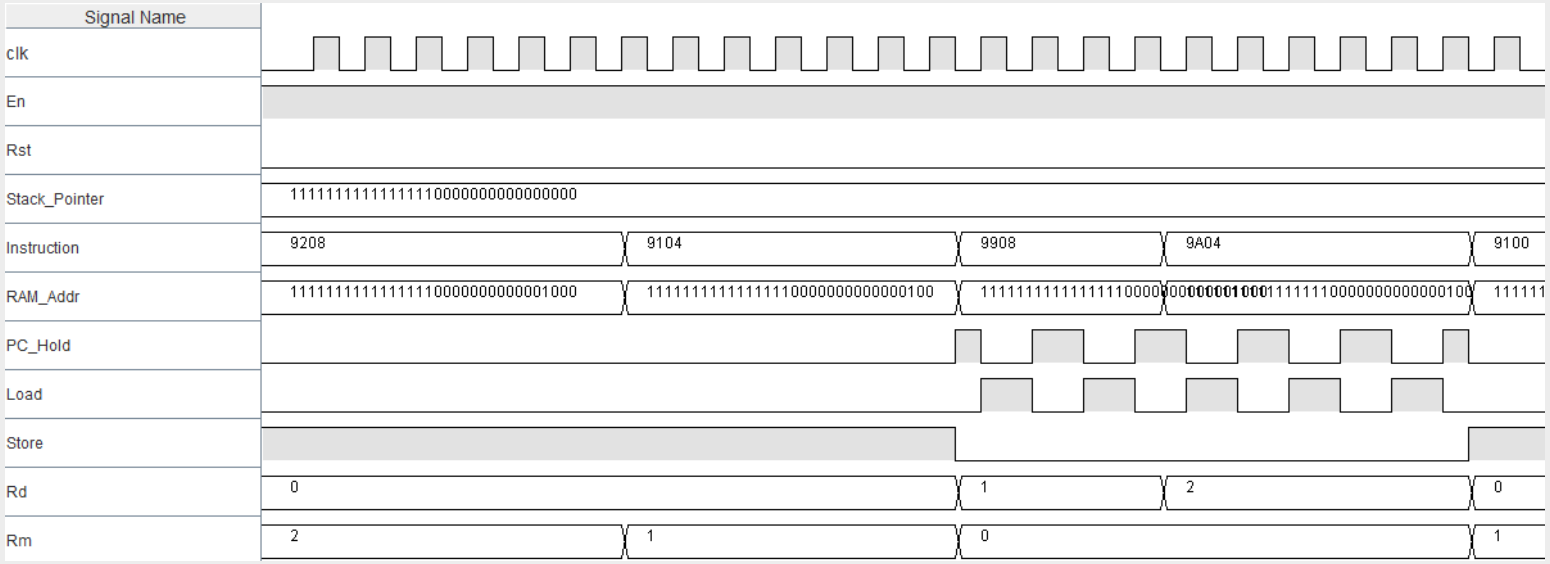
2

Le Controller


RAM
ROM
Banc de registres
Contrôleur
ALU
RAM_ADRESSE
ROM_ADRESSE


Instructions
Opcode_decoder
Shift_add_Sub_ Mov
Data_processing
Flags_APSR
ALU Instructions
Stack_ Pointer
Load_Store
Store/load To Selected Address
SP_Address
Hold The counter
Conditional
Counter
Rom_Address

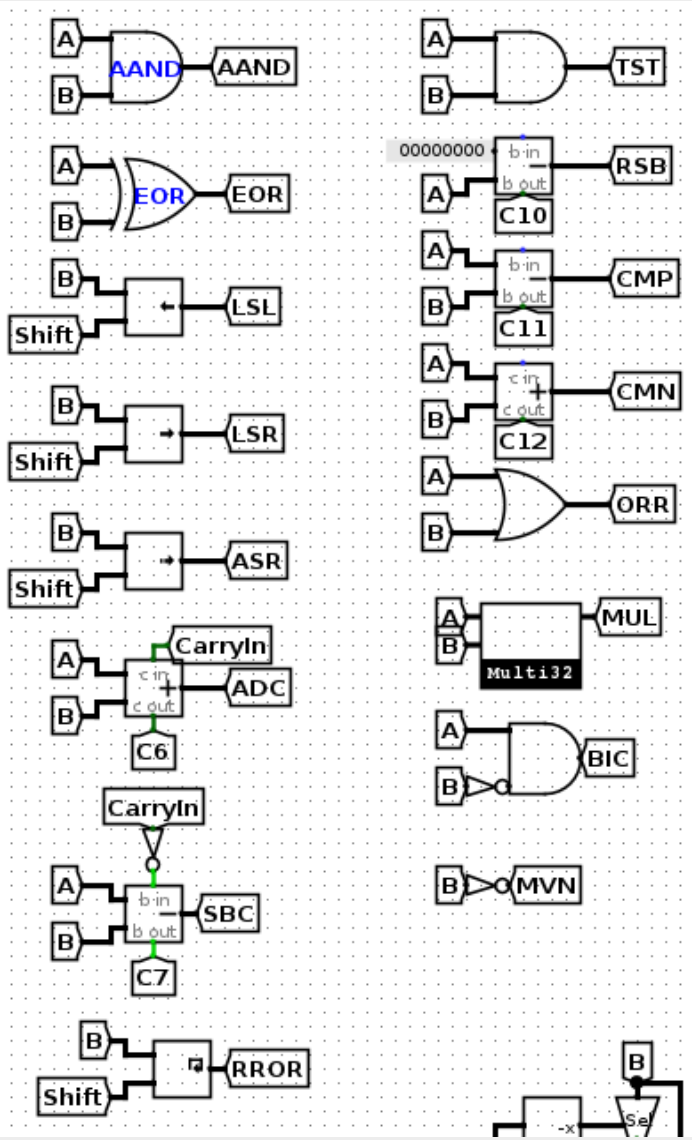
Load/Store

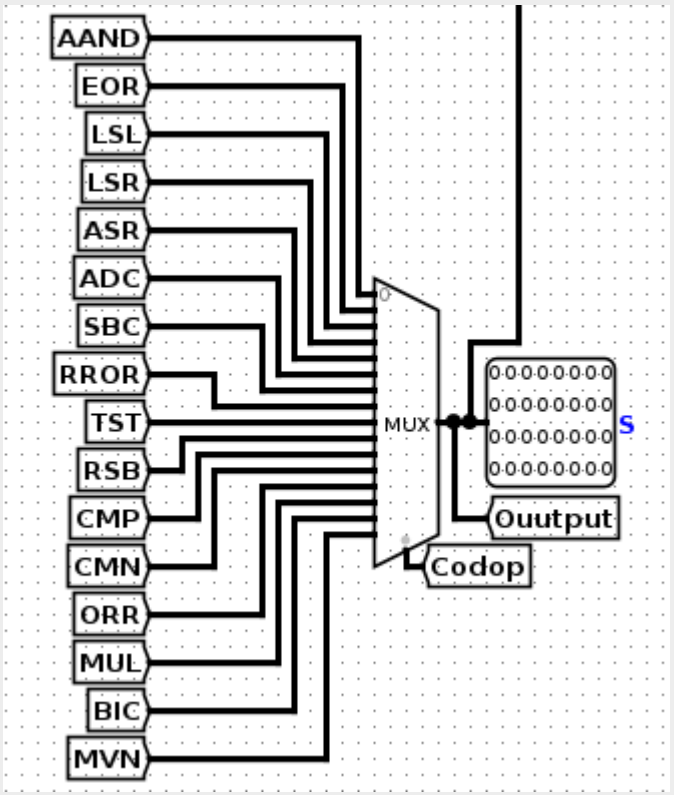

Signal Name
clk
En
Rst
Stack_Pointer
Instruction
RAM_Addr
PC_Hold
Load
Store
Rd
Rm
9208
9104
9908
9A04
9100
0
1
2
0
2
1
0
1

3

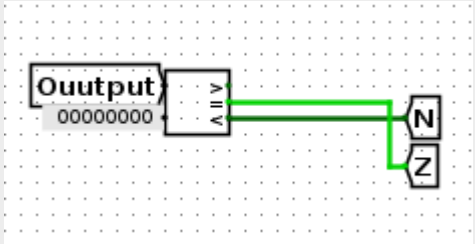
L'ALU

Les Opérations

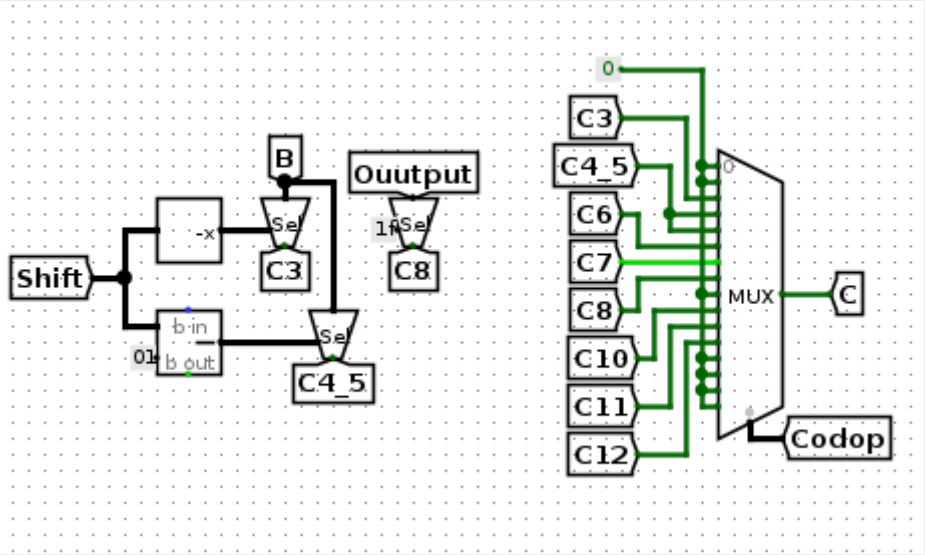




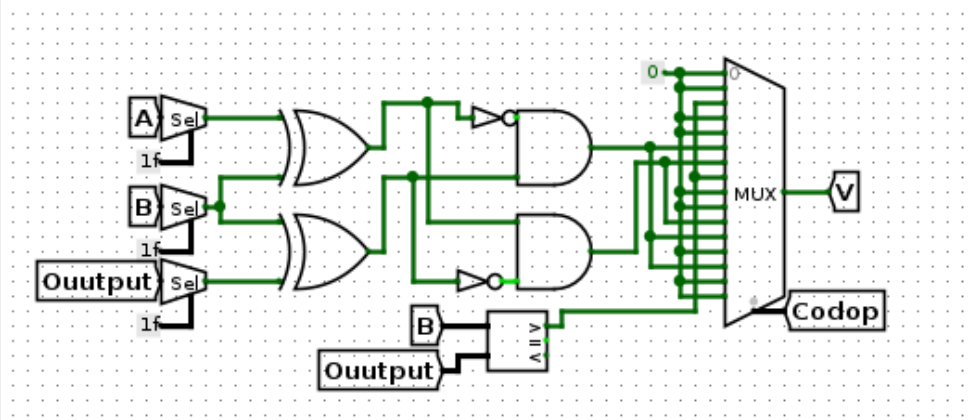
Les Flags



N & Z
(Négatif & Nul)



C
(Retenue sortante)



V
(Dépassement de capacité)

Tests

	Test des valeurs d'output									
			AND-0	EOR-1	LSL-2	LSR-3	ASR-4	ADC-5	SBC-6	ROR-7
A	B	Shift						CarryIn Off//On		
0	0		0	0				0//1	-1//0	
100	250		96	158				350//351	-151//-150	
65535	131071		65535	65536				196606//(+1)	-65537//(+1)	
1073741824	1073741824		1073741824	0				-2147483648	-1	
1	-1							0//1	1//2	
-100	-250							-350//-349	149//150	
	100	10			102400	0	0			102400
	2147483647	4			-16	134217727	134217727			-9

	Test des valeurs d'output								
A	B	TST-8	RSB-9	CMP-10	CMN-11	OR-12	MUL-13	BIC-14	MVN-15
0	0		0	0	0	0	0	0	-1
100	250		-100	-150	350	254	25000	4	-251
65535	131071		-65535	-65536	196606	131071	-196607	0	-131072
1073741824	1073741824		-1073741824	0	-2147483648	1073741824	0	0	-1073741825
1	-1		-1	2	0				
-100	-250		100	150	-350				

Résultat cohérent

Résultat non cohérent

	Test des valeurs de flag				Explication des valeurs :			V_C_Z_N		
			AND-0	EOR-1	LSL-2	LSR-3	ASR-4	ADC-5	SBC-6	ROR-7
A	B	Shift						CarryIn Off//On		
0	0		0_0_1_0	0_0_1_0				0_0_0//1_0	0_0_0//1_0	
100	250		0_0_0_0	0_0_0_0				0_0_0_0	0_1_0_1	
65535	131071		0_0_0_0	0_0_0_0				0_0_0_0	0_1_0_1	
1073741824	1073741824		0_0_0_0	0_0_1_0				1_0_0_1	0_1_0_1	
1	-1							0_1_1//0_0	0_1_0_0	
-100	-250							0_1_0_1	0_0_0_0	
	100	10			0_0_0_0	0_0_1_0	0_0_1_0			0_0_0_0
	2147483647	4			1_1_0_1	0_1_0_0	0_1_0_0			1_1_0_1

	Test des valeurs de flag				Explication des valeurs :			V_C_Z_N	
A	B	TST-8	RSB-9	CMP-10	CMN-11	OR-12	MUL-13	BIC-14	MVN-15
0	0	0_0_1_0	0_0_1_0	0_0_1_0	0_0_1_0	0_0_1_0	0_0_1_0	0_0_1_0	0_0_0_1
100	250	0_0_0_0	0_1_0_1	0_1_0_1	0_0_0_0	0_0_0_0	0_0_0_0	0_0_0_0	0_0_0_1
65535	131071	0_0_0_0	0_1_0_1	0_1_0_1	0_0_0_0	0_0_0_0	1_0_0_1	0_0_1_0	0_0_0_1
1073741824	1073741824	0_0_0_0	0_1_0_1	0_0_1_0	1_0_0_1	0_0_0_0	0_0_1_0	0_0_1_0	0_0_0_1
1	-1		0_1_0_1	0_1_0_0	0_1_1_0				
-100	-250		0_1_0_0	0_0_0_0	0_1_0_1				

V = Overflow

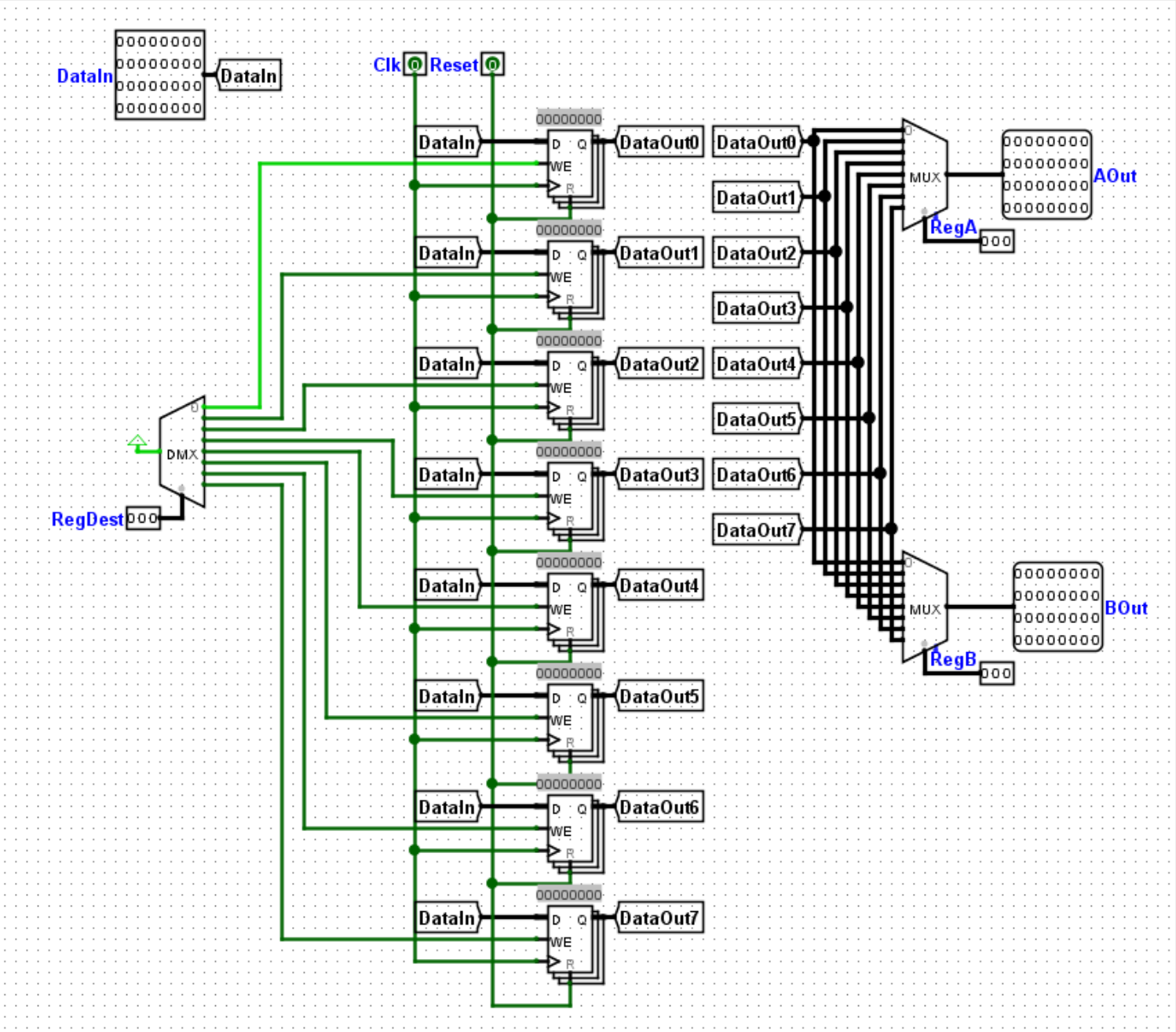
C = CarryOut

Z = Nul

N = Négatif

4

Le Banc de registre


DataIn
DataIn
Clk
Reset
DataIn
D Q
WE
R
DataOut0
DataOut1
DataOut2
DataOut3
DataOut4
DataOut5
DataOut6
DataOut7
DMX
RegDest
000
MUX
AOut
RegA
MUX
BOut
RegB

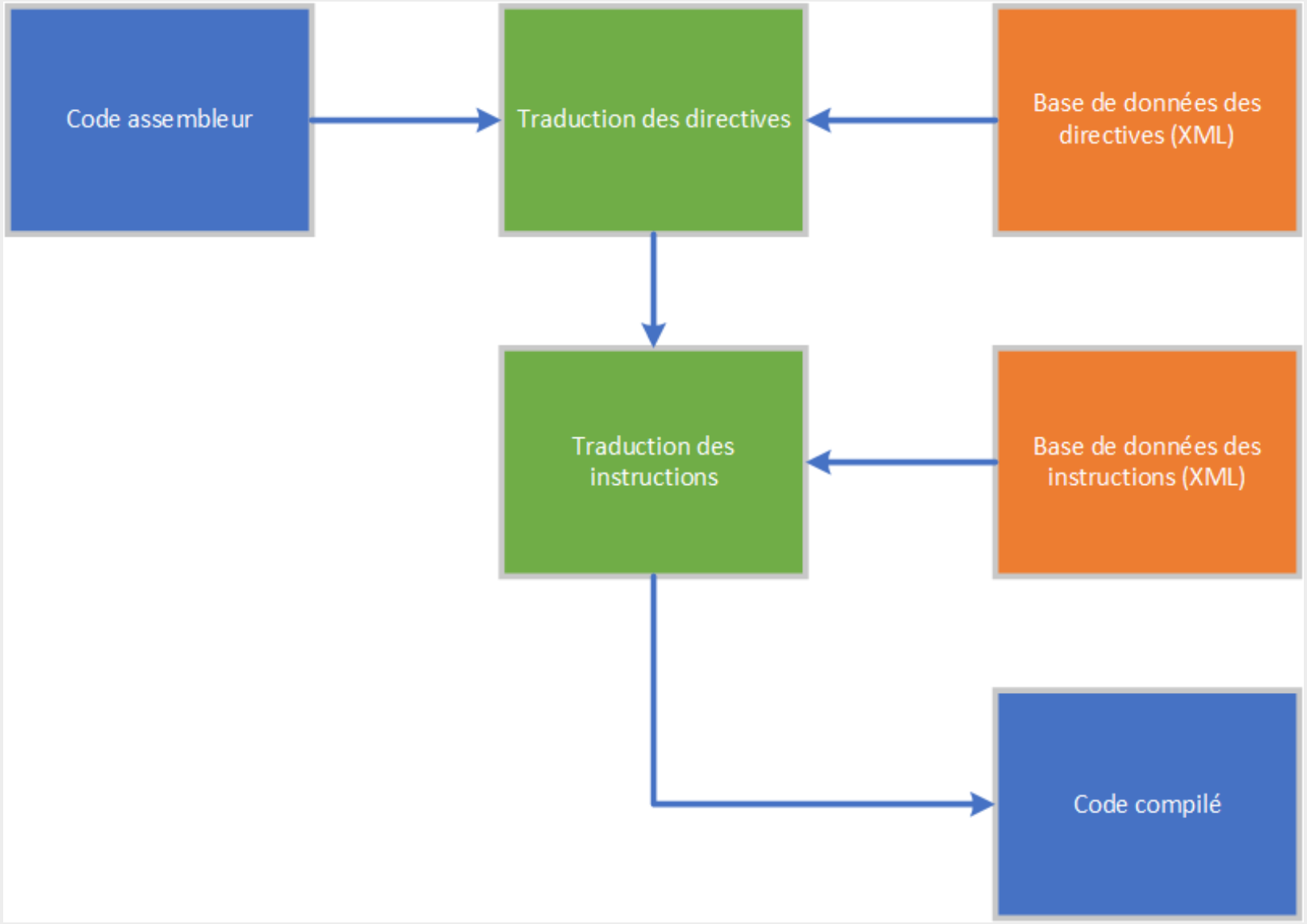
Tests

Entrées						Sorties obtenue										
DataIn	RegDest	Clock	Reset	RegA	RegB	Aout	Bout	R0	R1	R2	R3	R4	R5	R6	R7	
01000110	000	Rising Edge	0	000	000	01000110	01000110	01000110	01000110	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	000	Rising Edge	0	001	000	00000000	01000110	00000000	01000110	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	001	Rising Edge	0	001	000	01000110	00000000	01000110	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	001	Rising Edge	0	010	001	00000000	01000110	00000000	01000110	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	010	Rising Edge	1	010	001	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	010	Rising Edge	1	011	010	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	011	Rising Edge	1	011	010	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	011	Rising Edge	1	100	011	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	100	Falling Edge	1	100	011	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	100	Falling Edge	1	101	100	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	101	Falling Edge	1	101	100	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	101	Falling Edge	1	110	101	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	110	Falling Edge	0	110	101	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	110	Falling Edge	0	111	110	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	111	Falling Edge	0	111	110	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
01000110	111	Falling Edge	0	111	111	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	

	Résultat cohérent
	Résultat non cohérent

5

L’Assembleur


Code assembleur
Traduction des directives
Base de données des directives (XML)
Traduction des instructions
Base de données des instructions (XML)
Code compilé

6

Démonstration

Présentation des fonctionalités réalisées
par notre microprocesseur ARM

- Tests unitaires à la mains
- Circuits testeur
- Chronogramme

Opcode_decoder

Instruction en entrée	Résultat attendu		Résultat obtenu			
opcode	code sortie	note	code sortie	note	Statut	Observation
000000	10000	shift_add_sub_mov	10000	shift_add_sub_mov	Good	
001010	10000	shift_add_sub_mov	10000	shift_add_sub_mov	Good	
001111	10000	shift_add_sub_mov	10000	shift_add_sub_mov	Good	
010000	01000	Data_processing	01000	Data_processing	Good	
011000	00100	Load_Store	00100	Load_Store	Good	
100100	00100	Load_Store	00100	Load_Store	Good	
101100	00010	SP_Address	00010	SP_Address	Good	
110100	00001	Conditional	00001	Conditional	Good	

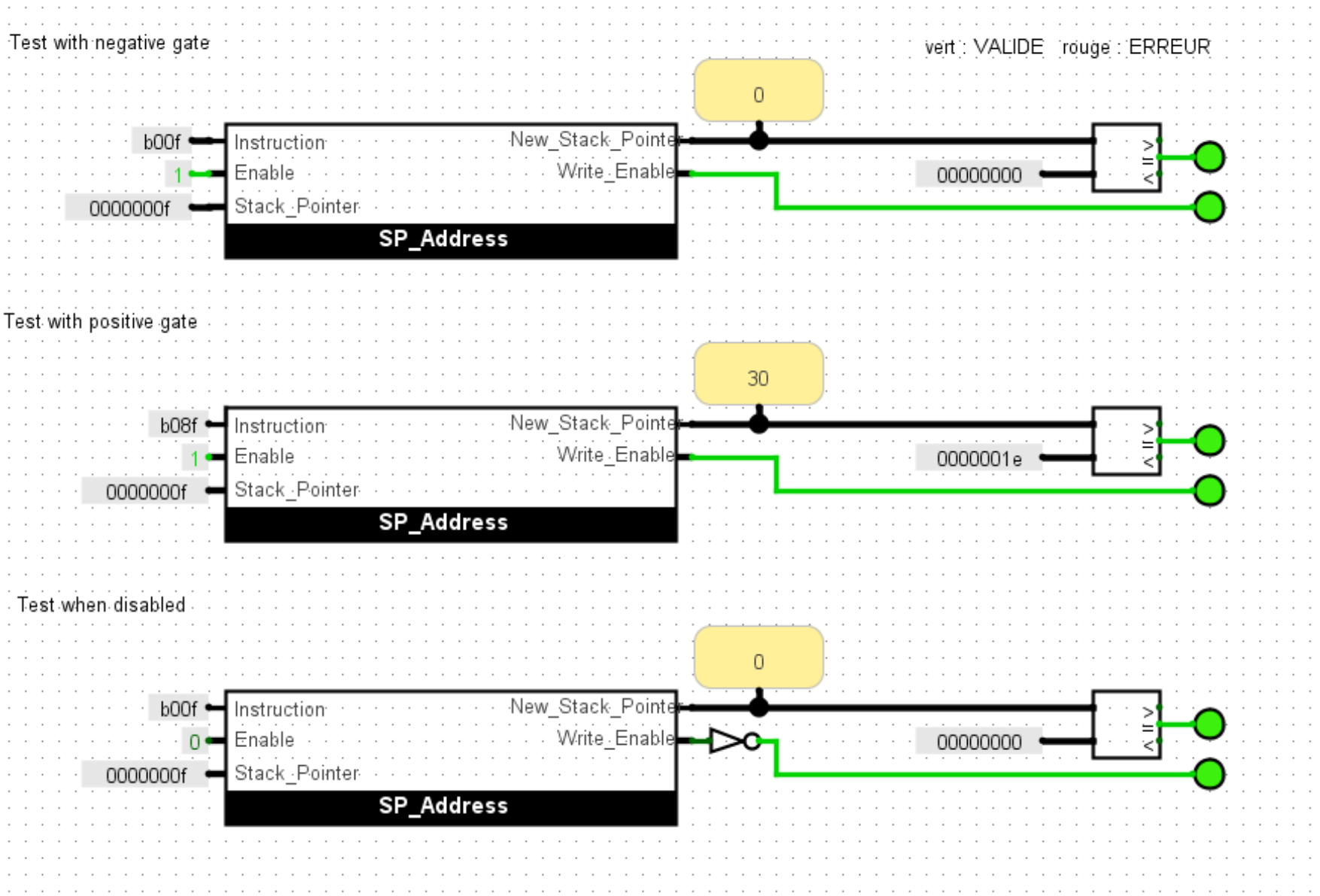
Shift_Add_Sub_Mov

Instruction en entrée		Résultat attendu									
Enable	Instruction	Carry	Imm32_En	Imm5	Imm32	ALU_Opcode	Rm	Rn	Rd	Flags	**note**
1	0000011010111001	0	0	11010	"00000000000000000000000000000000"	0100	000	111	001	1110	LSL
1	0000111010110100	0	0	11010	"00000000000000000000000000000000"	0011	000	110	100	1110	LSR
1	0001011000101111	0	0	11000	"00000000000000000000000000000000"	0100	000	101	111	1110	ASR
1	0001100110111001	0	0	00000	"00000000000000000000000000000000"	0101	110	111	001	1111	ADD
1	0001101010001100	1	0	00000	"00000000000000000000000000000000"	0110	010	001	100	1111	SUB
1	0001110110010001	0	1	00000	"00000000000000000000000000000110"	0101	000	000	001	1111	ADD_IMM
1	0001111001010110	1	1	00000	"00000000000000000000000000000001"	0110	000	000	1110	1111	SUB_IMM
1	0010000111010101	0	1	00000	"11111111111111111111111100101010"	1001	000	000	001	1100	MOV
0	0010000111000100	0	0	00000	"00000000000000000000000000000000"	0000	000	000	000	0000	DISABLE

Instruction en entrée		Résultat obtenu										Statut
Enable	Instruction	Carry	Imm32_En	Imm5	Imm32	ALU_Opcode	Rm	Rn	Rd	Flags	**note**	
1	0000011010111001	0	0	11010	"00000000000000000000000000000000"	0100	000	111	001	1110	LSL	Good
1	0000111010110100	0	0	11010	"00000000000000000000000000000000"	0011	000	110	100	1110	LSR	Good
1	0001011000101111	0	0	11000	"00000000000000000000000000000000"	0100	000	101	111	1110	ASR	Good
1	0001100110111001	0	0	00000	"00000000000000000000000000000000"	0101	110	111	001	1111	ADD	Good
1	0001101010001100	1	0	00000	"00000000000000000000000000000000"	0110	010	001	100	1111	SUB	Good
1	0001110110010001	0	1	00000	"00000000000000000000000000000110"	0101	000	000	001	1111	ADD_IMM	Good
1	0001111001010110	1	1	00000	"00000000000000000000000000000001"	0110	000	000	1110	1111	SUB_IMM	Good
1	0010000111010101	0	1	00000	"11111111111111111111111100101010"	1001	000	000	001	1100	MOV	Good
0	0010000111000100	0	0	00000	"00000000000000000000000000000000"	0000	000	000	000	0000	DISABLE	Good

Data_Processing

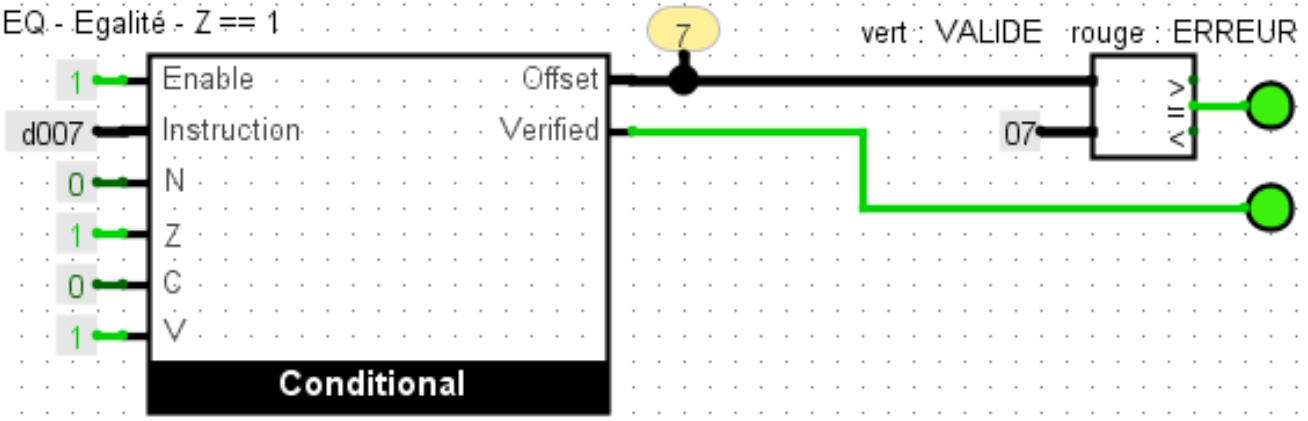
Instruction en entrée		Résultat attendu						Résultat obtenu						
Enable	Instructions	Opcode	Rn	Rm	Rd	Flags_Update	note	Opcode	Rn	Rm	Rd	Flags_Update	note	Statut
1	0100000000010001	0000	001	010	001	1110	AND	0000	001	010	001	1110	AND	Good
1	0100000001010001	0001	001	010	001	1110	EOR	0001	001	010	001	1110	EOR	Good
1	0100000010010001	0010	001	010	001	1110	LSL	0010	001	010	001	1110	LSL	Good
1	0100000011010001	0011	001	010	001	1110	LSR	0011	001	010	001	1110	LSR	Good
1	0100000100010001	0100	001	010	001	1110	ASR	0100	001	010	001	1110	ASR	Good
1	0100000101010001	0101	001	010	001	1111	ADC	0101	001	010	001	1111	ADC	Good
1	0100000110010001	0110	001	010	001	1111	SBC	0110	001	010	001	1111	SBC	Good
1	0100000111010001	0111	001	010	001	1100	ROR	0111	001	010	001	1100	ROR	Good
1	0100001000010001	1000	001	010	001	1110	TST	1000	001	010	001	1110	TST	Good
1	0100001001010001	1001	000	010	001	1111	RSB	1001	000	010	001	1111	RSB	Good
1	0100001010010001	1010	001	010	001	1100	CMP	1010	001	010	001	1100	CMP	Good
1	0100001011010001	1011	001	010	001	1111	CMN	1011	001	010	001	1111	CMN	Good
1	0100001100010001	1100	001	010	001	1110	ORR	1100	001	010	001	1110	ORR	Good
1	0100001101010001	1101	001	010	001	1100	MUL	1101	001	010	001	1100	MUL	Good
1	0100001110010001	1110	001	010	001	1110	BIC	1110	001	010	001	1110	BIC	Good
1	0100001111010001	1111	000	010	001	1110	MVN	1111	000	010	001	1110	MVN	Good
0	0100001111010001	0000	000	000	000	0000	DISABLE	0000	000	000	000	0000	DISABLE	Good


Test with negative gate
vert : VALIDE rouge : ERREUR
0
b00f
Instruction
New_Stack_Pointer
1
Enable
Write_Enable
0000000f
Stack_Pointer
SP_Address
00000000
Test with positive gate
30
b08f
Instruction
New_Stack_Pointer
1
Enable
Write_Enable
0000000f
Stack_Pointer
SP_Address
0000001e
Test when disabled
0
b00f
Instruction
New_Stack_Pointer
0
Enable
Write_Enable
0000000f
Stack_Pointer
SP_Address
00000000

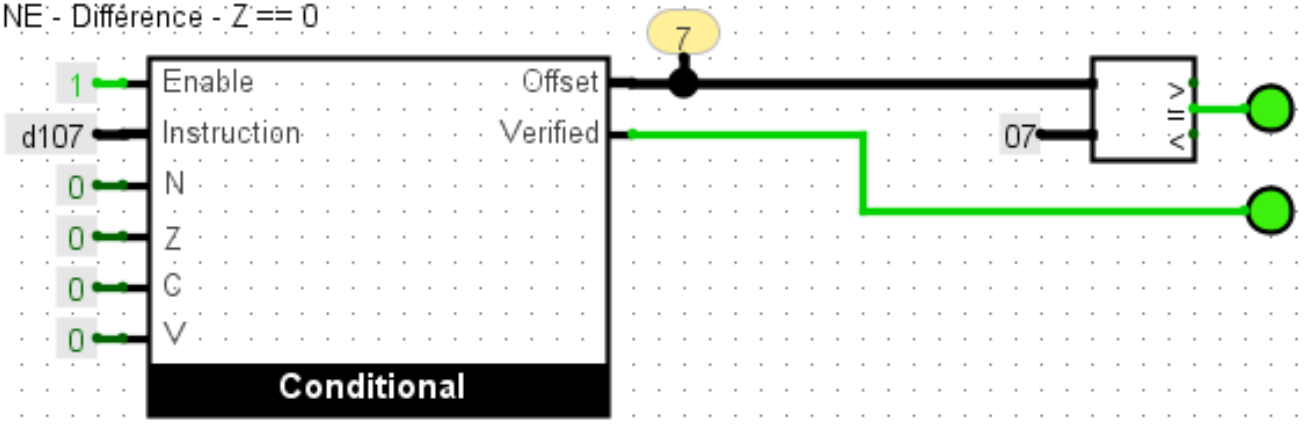
N : Résultat négatif, égal au bit de pods fort au résultat
C : Retenue
Z : Résultat nul, égal à 1 si le résultat est 0
V : Dépassement de capacité

code : 0000 - EQ - Egalité - Z == 1

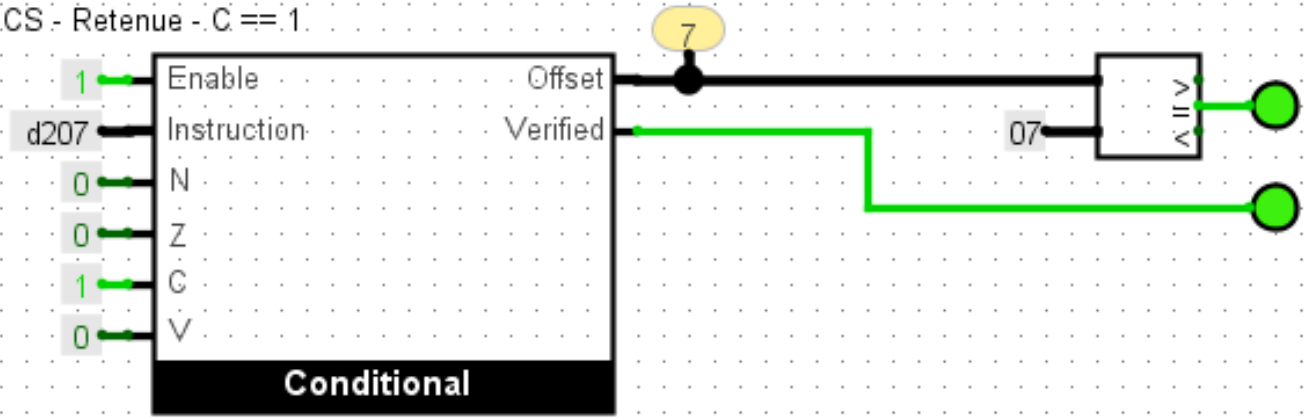
vert : VALIDE rouge : ERREUR



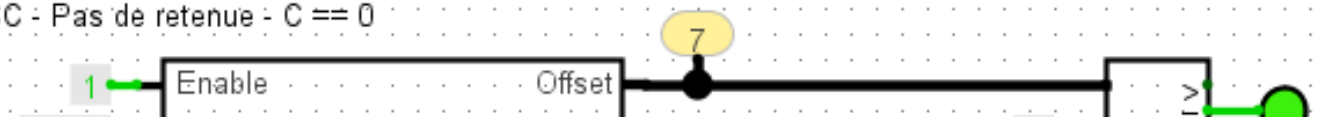
code : 0001 - NE - Différence - Z == 0



code : 0010 - CS - Retenue - C == 1



code : 0011 - CC - Pas de retenue - C == 0



Flags_APSR
Chronogramme