

Exercice 1. Numération

Combien de bits sont nécessaires pour écrire le nombre 1026 ?

Donnez la valeur en hexadécimal et binaire.

Exercice 2

Soit le programme en langage assembleur ci-dessous

```
movs r0, #10
compteur:
    subs r0, r0, #1
    cmp r0, #5
    bne compteur
```

- 1) Expliquer le fonctionnement de ce programme
- 2) Après compilation, la fenêtre de désassemblage donne le résultat ci-dessous

216		movs r0, #10
080001ec:	0x0000 ????	movs r0, #10
218		subs r0, r0, #1
080001ee:	0x0000 ????	subs r0, #1
219		cmp r0, #5
080001f0:	0x0000 ????	cmp r0, #5
220		bne compteur
080001f2:	0x0000fcd1	bne.n 0x80001ee <Reset_Handler+114>

Grâce aux documents ci-dessous, donnez les valeurs des ‘?’.

Justifier l’encodage de 0xFCD1 pour la fonction bne compteur. Pour information .n signifie négatif !

A6.7.75 MOV (immediate)

Move (immediate) writes an immediate value to the destination register. It can optionally update the condition flags based on the value.

Encoding T1 All versions of the Thumb ISA.

MOV<S> <Rd>, #<imm8>

Outside IT block.

MOV<C> <Rd>, #<imm8>

Inside IT block.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	Rd			imm8							

d = UInt(Rd); setflags = !InITBlock(); imm32 = ZeroExtend(imm8, 32); carry = APSR.C;

A6.7.132 SUB (immediate)

This instruction subtracts an immediate value from a register value, and writes the result to the destination register. It can optionally update the condition flags based on the result.

Encoding T1 All versions of the Thumb ISA.

SUBS <Rd>, <Rn>, #<imm3>

Outside IT block.

SUB<C> <Rd>, <Rn>, #<imm3>

Inside IT block.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0 0 0			1 1		1	1	imm3			Rn			Rd		

d = UInt(Rd); n = UInt(Rn); setflags = !InITBlock(); imm32 = ZeroExtend(imm3, 32);

Encoding T2 All versions of the Thumb ISA.

SUBS <Rdn>, #<imm8>

Outside IT block.

SUB<C> <Rdn>, #<imm8>

Inside IT block.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	1	Rdn			imm8							

d = UInt(Rdn); n = UInt(Rdn); setflags = !InITBlock(); imm32 = ZeroExtend(imm8, 32);

A6.7.27 CMP (immediate)

Compare (immediate) subtracts an immediate value from a register value. It updates the condition flags based on the result, and discards the result.

Encoding T1 All versions of the Thumb ISA.

CMP<C> <Rn>, #<imm8>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	1	Rn			imm8							

n = UInt(Rdn); imm32 = ZeroExtend(imm8, 32);

A6.7.12 B

Branch causes a branch to a target address.

Encoding T1 All versions of the Thumb ISA.

B<C> <label>

Not allowed in IT block.

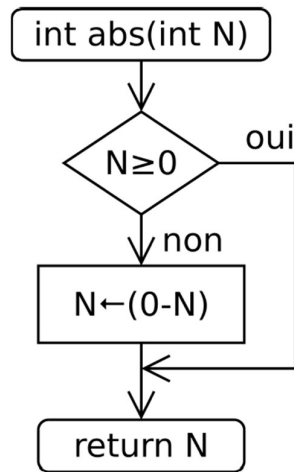
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	cond				imm8							

```
if cond == '1110' then UNDEFINED;  
if cond == '1111' then SEE SVC;  
imm32 = SignExtend(imm8:'0', 32);  
if InITBlock() then UNPREDICTABLE;
```

cond	Mnemonic extension	Meaning (integer)	Meaning (floating-point) ^{ab}	Condition flags
0000	EQ	Equal	Equal	Z == 1
0001	NE	Not equal	Not equal, or unordered	Z == 0
0010	CS ^c	Carry set	Greater than, equal, or unordered	C == 1
0011	CC ^d	Carry clear	Less than	C == 0
0100	MI	Minus, negative	Less than	N == 1
0101	PL	Plus, positive or zero	Greater than, equal, or unordered	N == 0
0110	VS	Overflow	Unordered	V == 1
0111	VC	No overflow	Not unordered	V == 0
1000	HI	Unsigned higher	Greater than, or unordered	C == 1 and Z == 0
1001	LS	Unsigned lower or same	Less than or equal	C == 0 or Z == 1
1010	GE	Signed greater than or equal	Greater than or equal	N == V
1011	LT	Signed less than	Less than, or unordered	N != V
1100	GT	Signed greater than	Greater than	Z == 0 and N == V
1101	LE	Signed less than or equal	Less than, equal, or unordered	Z == 1 or N != V
1110	None (AL) ^e	Always (unconditional)	Always (unconditional)	Any

Exercise 3

Faire un programme qui calcule la valeur absolue d'un nombre entier en suivant l'algorithme suivant. Le nombre N sera enregistré dans le registre r0 et le résultat donné dans r0. Le programme est court, l'objectif de l'exercice est de trouver les instructions dans le programming manual.



Exercice 4

Faire un programme en assembleur où un entier n positif est dans le registre R0 et et calcule le factoriel de n (n!). n sera codé au maximum sur 8 bits. n est donné est dans le registre r0 et le résultat sera également dans r0.