



## Systèmes Embarqués 1 & 2

Classes T-2/I-2 // 2017-2018

### a.03 – Jeu d'instructions du $\mu$ P ARM

#### Exercices

#### Exercice 1

Transcrire en assembleur les algorithmes C ci-dessous

(a) 

```
int i = 0;
int j = 0;
for (i=0; i<100; i++) j += i;
```

(b) 

```
int i = 100;
unsigned int j = 2;
while (i>0) {
    j *= j; i--;
}
```

(c) 

```
int i = 8;
unsigned int j = 0x7000;
do {
    j = j >> 1; i--;
} while (i > 0);
```

(d) 

```
int i;
int j = 0;
switch(i) {
    case 0: j=11; break;
    case 1: j=8; break;
    case 2: j=25; break;
    case 3: j=99; break;
    case 4: j=33; break;
    default: j=-1; break;
}
```

(e) 

```
int i;
int j=0;
if (i>0) j = i % 16;
else if (i==0) j = 0xaa;
else j = -i / 16;
```

**Exercice 2**

Coder en assembleur ARM l'algorithme ci-dessous

```
short toto;    // 2 octets
char i;        // 1 octet

toto = 20;
i = 0;
while (i<7) { toto += toto; toto -= i; i++; }
```

**Exercice 3**

Coder en assembleur ARM l'instruction ci-dessous

```
k := 0;
switch (i) {
  case 0 : k := -20; break;
  case 2 : k := -2;  break;
  case 4 : k := 0;   break;
  case 5 : k := 5;   break;
  case 8 : k := 24;  break;
}
```

**Exercice 4**

Coder en assembleur ARM l'algorithme calculant la parité verticale d'un certain nombre d'octets de données.

Code en C de la fonction

```
#define LENGTH 4
char chaine[LENGTH+1] = {0xC4,0xBF,0x1F,0xAD,0x00};
main() {
  char n = 0;
  while (n < LENGTH) {
    chaine[LENGTH] = chaine[LENGTH] ^ chaine[n] ;
    n++;
  }
}
```

Exemple

1	1	0	0	0	1	0	0
1	0	1	1	1	1	1	1
0	0	0	1	1	1	1	1
1	0	1	0	1	1	0	1
1	1	0	0	1	0	0	1

**Exercice 5**

Pour le code assembleur ci-dessous

```
res:    .short    1
var1:   .byte     67

main:   ldr        r0, =var1
        ldrb       r4, [r0]
        lsl        r2, r4, #3
        lsl        r4, #1
        add        r4, r2
        ldr        r0, =res
        strh       r4, [r0]
```

- (a) Quel sera la valeur stockée dans la variable « res » une fois que le code se sera déroulé?
- (b) Expliquer en quelques lignes la fonction de ce code.
- (c) Quelle est la valeur résultante dans le registre R2 (en décimal, hexa et binaire) ?
- (d) Quelle est la valeur résultante dans le registre R4 (en décimal, hexa et binaire) ?
- (e) Pour quelle fourchette de valeur (valeur minimale et maximale) de la variable « var1 », cet algorithme est-il valable ?