TD 3 Architecture logicielle
Accès mémoire

Dans tout le TD, on considère le jeu d'instructions DIDE.

3.1 Format des instructions

– Donner le codage hexadécimal de l'instruction LOAD R4, 16(R6).

3.2 Etude des instructions

Initialement, le registre R0 contient 0, le registre R1 contient 0x95842103, le registre R2 contient 0x00001000 ; le contenu de la mémoire est 0 dans toutes les adresses, sauf celles décrites table. 1.

Adresse	Contenu
0x00001000	0x12
0x00001001	0x34
0x00001002	0x56
0x00001003	0x78
0x00001004	0x9A
0x00001005	0xBC
0x00001006	0xDE
0x00001007	0xF0

TAB. 1 – Contenu de la mémoire

– En repartant à chaque fois de l'état initial, donner l'état des registres et des cases mémoires modifiés après l'exécution des instructions :
  a) LOAD R1 0(R2) b) STORE R1 4(R2)

3.3 Plan mémoire

La déclaration en C de la fig. 1 définit une occupation mémoire ; les valeurs sont notées en hexadécimal en commentaire. Le placement est aligné et en big endian. La structure est implantée à partir de l'adresse 0x00002000.

– Donner le contenu des octets mémoire concernés, en supposant que la structure *data* est implantée à partir de l'adresse 0.

```
struct     {
           int          a;       /*x'11121314'*/
           double       b;       /*x'2122232425262728'*/
           char*        c;       /*x'31323334'*/
           char         d[7];    /*'A', 'B', 'C', 'D', 'E', 'F', 'G'*/
           short        e;       /*x'5152'*/
           int          f;       /*x'61626364'*/
           } data;
```

FIG. 1 – Déclaration de variables en LHN

3.4 Programmation

On considère le plan mémoire de la table 2.

Adresses	Contenu
0x00001000 - 0x00001003	x
0x00001004 - 0x00001007	t
0x00001008 - 0x0000100B	z
0x0000100C - 0x0000100F	M[0]
0x00001010 - 0x00001013	M[1]
0x00001014 - 0x00001017	M[2]

TAB. 2 – Plan mémoire

- Ecrire une séquence d'instructions qui correspond à l'affectation x := x+2*z.
- Ecrire une séquence d'instructions qui correspond au fragment de programme suivant, où toutes les variables sont entières :

  z := M[0]+ x ;
  t := M[2] + x ;
  x := M[1] + z + t ;