Architecture Elec3A

Examen (Durée : 1h) ; *Cours, TDs. et TPs. autorisés*

Exercice I (Question de cours):

(1) Lesquelles, des constantes suivantes, peuvent être chargées dans un registre en utilisant une instruction *MOV*, comme : *MOV R0,#____* (valeur immédiate) ou bien *MOV R0,#__,__* (valeur immédiate avec une rotation à droite d'un nombre pair entre 0 et 30) ?

a) 192 ; b) 259 **c)** 768

(2) Toutes les lignes de code assembleur ARM ci-dessous contiennent des erreurs. Les erreurs peuvent être de différents genres : erreur de syntaxe, adressage incorrect, utilisation de registres inappropriée, des constantes invalides, etc. Identifier dans chaque instruction où apparaît l'erreur et indiquer la nature de l'erreur présente.

a) MOV R4, #513 ; **b)** ADD R0, R0, 1 ; c) AND R5, R6;

(3) Dans chacune des questions ci-dessous, vous devez multiplier le contenu du registre R0 par une constante sous forme d'une seule instruction sans utilisation d'autres registres et sans utilisation d'instruction du type MUL ou MLA.

Q1 : R0 :=R0 x 5 ***ADD***.....

Q2 : R0 :=R0 x 8 ***MOV*** .....

Q3 : R0 :=R0 x (-7) ***SUB***.....

Exercice II (Assembleur ARM) :

Ecrire un sous-programme en assembleur ARM7 nécessaire pour la fonction ci-dessous. Ecrire ce sous-programme de telle sorte qu'il puisse être appelé depuis un programme en C. Dans ce cas, les variables ***i, x, y*** doivent être internes à cette fonction, donc doivent être dans des registres. Bien utiliser les registres correspondants au passage des paramètres (**a**, **b**, **c**) à la fonction **f**. Donc, **a** sera reçu dans R0, **b** sera reçu dans r1 et **c** sera reçu dans r2.

```
f ( int a , b , c )
    {
        int  i , x , y ;
             y = 0;
             if ( a = = b)
                  x = 3;
             else
                  x = 5;
             for ( i = 1 ; i < x ; i ++)
                                y = c + y;
             return  y ;
        }
```

Architecture Elec3A

Examen (Durée : 1h) ; *Cours, TDs. et TPs. autorisés*

Exercice I (Question de cours):

(1) Lesquelles, des constantes suivantes, peuvent être chargées dans un registre en utilisant une instruction *MOV*, comme : *MOV R0,#____* (valeur immédiate) ou bien *MOV R0,#__,__* (valeur immédiate avec une rotation à droite d'un nombre pair entre 0 et 30) ?

a) 192 ; b) 259 **c)** 768

(2) Toutes les lignes de code assembleur ARM ci-dessous contiennent des erreurs. Les erreurs peuvent être de différents genres : erreur de syntaxe, adressage incorrect, utilisation de registres inappropriée, des constantes invalides, etc. Identifier dans chaque instruction où apparaît l'erreur et indiquer la nature de l'erreur présente.

a) MOV R4, #513 ; **b)** ADD R0, R0, 1 ; c) AND R5, R6;

(3) Dans chacune des questions ci-dessous, vous devez multiplier le contenu du registre R0 par une constante sous forme d'une seule instruction sans utilisation d'autres registres et sans utilisation d'instruction du type MUL ou MLA.

Q1 : R0 :=R0 x 5 ***ADD***.....

Q2 : R0 :=R0 x 8 ***MOV*** .....

Q3 : R0 :=R0 x (-7) ***SUB***.....

Exercice II (Assembleur ARM) :

Ecrire un sous-programme en assembleur ARM7 nécessaire pour la fonction ci-dessous. Ecrire ce sous-programme de telle sorte qu'il puisse être appelé depuis un programme en C. Dans ce cas, les variables ***i, x, y*** doivent être internes à cette fonction, donc doivent être dans des registres. Bien utiliser les registres correspondants au passage des paramètres (**a**, **b**, **c**) à la fonction **f**. Donc, **a** sera reçu dans R0, **b** sera reçu dans r1 et **c** sera reçu dans r2.

```
f ( int a , b , c )
      {
            int  i , x , y ;
                  y = 0;
                  if ( a = = b)
                        x = 3;
                  else
                        x = 5;
                  for ( i = 1 ; i < x ; i ++)
                                    y = c + y;
                  return  y ;
            }
```

UNIVERSITE DE BOURGOGNE DEPARTEMENT I.E.M.	L2, Elec3A Session 1, Decembre2021, 1h
Ordinateurs, Portable et connexion à internet interdits. Cours, TD et TP autorisés	

Numéro Anonymat :

Problème n° 1 : (5 points)

Le contrôle de bit est une fonction qui très utilisée en électronique lors des communications des données d'un point à un autre. Le schéma électronique ci-contre joue cette fonction. *n* représente le nombre de bit à transmettre en parallèle. EMISSION A représenté le site de départ des données et EMISSION B représente le sire de réception des données situées à une distance quelconque.

<u>Définition de la parité en logique</u> : "Cela consiste à ajouter un bit 1 à l'information utile pour que le nombre total de bits à 1 soit paire. Dans le cas contraire, on parle d'imparité".

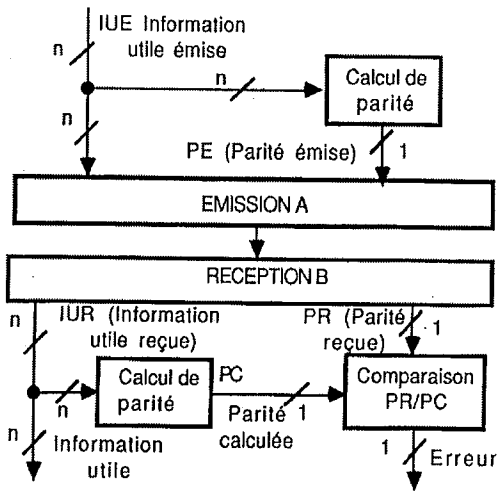


Figure 1: Schéma fonctionnel

1) Compléter la table de vérité ci-après en définissant les valeurs prises par les fonctions P (parité) et I (imparité), selon la définition ci-dessus :

X	Y	Z	P	I
0	0	0	0	1
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

2) Remplir les tableaux de Karnaugh ci-après pour la fonction P paire.

P	00	01	11	10
0				
1				

3) Déduire la fonction P pour la parité

P=

4) Remplir les tableaux de Karnaugh ci-après pour la fonction I impaire.

I	00	01	11	10
0				
1				

5) Déduire la fonction I pour l'imparité

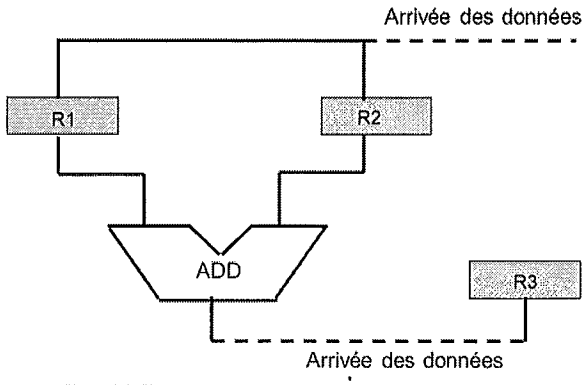
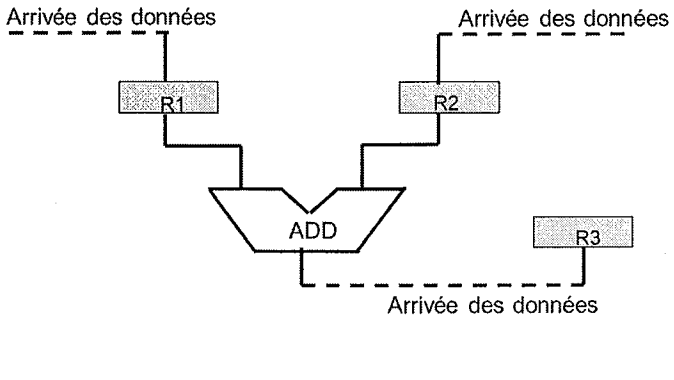
I=

6) On réalise le logigramme de la fonction **P** simplement à l'aide des portes suivantes :
- Nor

Donner le logigramme en portes Nor

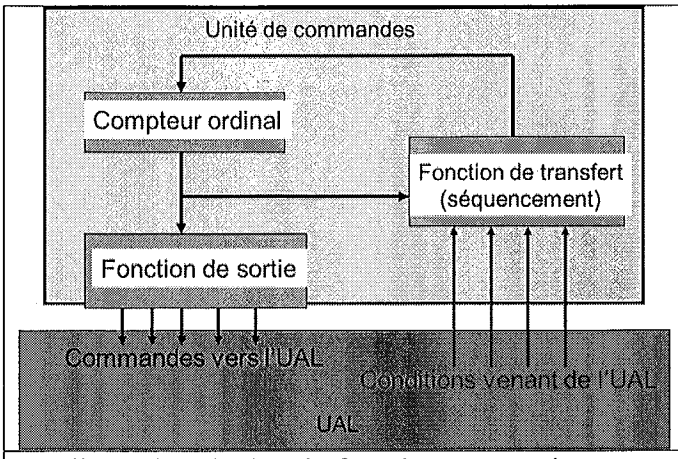
Problème 2 : (5 points)

1) On présente deux modèles d'une opération d'addition. Dans un système complexe. Dans chaque cas expliquer le processus et définir le nombre de top d'horloges nécessaires pour accomplir l'opération

	
Expliquer le fonctionnement :	Expliquer le fonctionnement :

Donner le nombre de top d'horloge :	Donner le nombre de top d'horloge :

2) On le schéma fonctionnel d'un système complexe



A qoui sert le bloc commande ? :

A qoui sert le bloc UAL ?

Expliquer le principe de fonctionnement du compteur Ordinal :

Donner la définition et le rôle des périphériques ci-dessous

Registre :

Bus :

Adresse :