centre de télé-enseignement universitaire formations à distance à votre rythme par e-learning

ctu UNIVERSITÉ DE FRANCHE-COMTÉ UFC

Licence d'informatique - Année 2018/2019

Architecture des Ordinateurs

Examen de Aout 2019 - Durée 3 heures

Documents autorisés : aucun - pas de calculatrice

- Si vous détectez une erreur de sujet, merci de l'indiquer dans votre copie.
- Pour être valables, vos réponses doivent être justifiées sur votre copie.
- Complétez et joignez la feuille de réponse à votre copie
- Utilisez les noms des variables et des fonctions proposées dans l'énoncé.
- Les questions sont indépendantes.

Exercice 1 : Représentation des nombres

```
1  int foo(int x,int y){
2        int var = x+y, res = var >> 1;
3        if ((var & 1)== 1) res = res + 1;
4        return res; }
5  int foo2(int x,int y){ return (x | y) - ((x^y)>>>1); }
6  void main(){
7       int N1= 0x00FEDCBA,N2=0xABCDEF00;
8       int N3 = Float.floatToIntBits(-11.75f);
9      //C pour N3 x=-11.25f ; unsigned N3= *(unsigned*) & x;
10     printTabCharZero(toHexString(N3));
11     printTabCharZero(toHexString(N1&N2));
12     printTabCharZero(toHexString(N1|N2));
13     printTabCharZero(toHexString(N1^N2));
14     printTabCharZero(toHexString(N2>>1));
15     printTabCharZero(toHexString(N2>>>1)); // C (unsigned)N2 >>1
16     printTabCharZero(toString(foo(4,7)));
17     printTabCharZero(toHexString(foo(0x80000000,0x80000000)));
18     printTabCharZero(toString(foo2(4,7)));
19     printTabCharZero(toHexString(foo2(0x80000000,0x80000000)));}
```

Question 1 : Expliquez et donnez, sur la feuille de réponse, les valeurs hexadécimales affichées par les lignes 10 à 15 du programme ci-dessus

Question 2 : Expliquez et donnez, sur la feuille de réponse, la valeur hexadécimale affichée par les lignes 16 à 19 du programme ci-dessus. (Attention : les paramètres sont des valeurs signées représentées sur 32 bits). Quel est l'intérêt de la fonction *foo2* ?

Exercice 2 : Assembleur Mips

Question 1 : Passage des paramètres par les registres Complétez, sur la feuille de réponse, le code de la fonction *foo* de l'exercice précédent et de son appel avec les valeurs 4 et 7 . Le code affichera le résultat en décimal.

Question 2 : Passage des paramètres par la pile Complétez, sur la feuille de réponse, le code de la fonction *foo* et de son appel avec les valeurs 4 et 7 et l'adresse de la variable résultat **0x**2000.

```
  # Programme pour les questions
      suivantes
.text 0
   addi  $a0, $0,0x2000
   addi  $a1, $0,2
test:
   lw  $t0,0( $a0)
   lw  $t1,4( $a0)
   and  $t0, $t0, $t1 # inst 1
   sw  $t0,8( $a0) # inst 2
   addi  $a0, $a0,16
   addi  $a1, $a1,-1
   bne  $a1, $0,test # inst 3
```

Exercice 3 : Hiérarchie mémoire

Soit un cache pédagogique de 128 octets, organisé par blocs de 16 octets puis de 32 octets. Nous souhaitons évaluer les performances du programme ci-dessus. Nous évaluerons la fonction de correspondance directe.

Question 1 : Que fait le programme ci-dessus ? Combien d'itérations sont effectuées ?

Question 2 : Expliquez et donnez, sur la feuille de réponse, la taille en bit de l'étiquette, de l'index et du déplacement dans les deux cas.

Question 3 : Expliquez et donnez, sur la feuille de réponse, les taux de succès lors de l'accès au cache. Pour cela, complétez les valeurs hexadécimales des trois champs et du succès d'accès au cache (S ou D) pour les deux premières itérations.

Exercice 4 : Pipeline

Question 1 : Indiquez les dépendances lecture après écriture du programme ci-dessus.

Question 2 : Expliquez ce qu'est le branchement retardé d'un cycle.

Question 3 : Complétez, sur la feuille de réponse, le diagramme temporel.

Question 4 : Réorganisez, sur la feuille de réponse, le programme pour ne pas avoir de cycle d'attente avec l'envoi et le branchement retardé d'un cycle.

Exercice 5 : Architecture

Question 1 : Expliquez et donnez, sur la feuille de réponse, les valeurs hexadécimales représentant les instructions 1 à 3 du programme ci-dessus.

Question 2 : Expliquez et donnez, sur la feuille de réponse, les valeurs numériques des signaux lors du traitement des trois instructions précédentes à la première itération du programme. La mémoire **0x**2000 contient la valeur **0x**1 , **0x**2004 la valeur **0x**2.

Exercice 6 : Exceptions et Entrées/sorties

Question 1 : Expliquez ce qui pose problème lors de l'exécution des instructions suivantes :

```
1    lui  $t1,0x7FFF
2    ori  $t1, $t1,0xFFFF
3    addi  $t1, $t1,1
```

Question 2 : Corrigez le programme permettant de lire et d'afficher un caractère sur la console externe.

```
1  clavier:
2    sw  $t0,0x7F04( $0)
3    bne  $t0, $0,clavier
4    lw  $t0,0x7F00( $0)
5    lw  $t0,0x7F08( $0)
```

Feuille de réponse à joindre à votre copie sans nom ni numéro d'étudiant

toHexString(N3)	
toHexString(N1& N2)	
toHexString(N1 | N2)	
toHexString(N1 ^ N2)	
toHexString(N2 >> 1)	
toHexString(N2>>> 1)	
toString(foo(4,7))	
toString(foo2(4,7))	
toHexString(foo(0x80000000,0x80000000))	
toHexString(foo2(0x80000000,0x80000000))	

```
    addi  $a0,_______,_______
    addi  $a1,_______,_______
    jal  foo
    add  $a0,_______,_______
    addi  $v0,_______,_______
    syscall
    addi  $v0,_______,_______
    syscall
foo:
#  $a0  et  $a1  paramètres
#  $v0  résultat
    sw  $t0,_______($sp)
    add  $t0, $a0,_______
    sra  $v0,_______,_______
    andi  $t0,_______,_______
    beq  $t0,_____,_____________
    addi  $v0,_______,_______
foo_fin:
    lw  $t0,_______($sp)
    jr _______
```

```
  # passage par la pile
    sw  $t0,_______($sp)
    addi  $t0, $0,4
    sw  $t0,_______($sp)
    addi  $t0, $0,7
    sw  $t0,_______($sp)
    addi  $t0, $0,0x2000
    sw  $t0,_______($sp)
    addi  $sp, $sp,_______
    jal  foo_pile
    addi  $sp, $sp,_______
    lw  $a0,_______($0)
  # idem foo
foo_pile:
    sw  $t0,0($sp)
    sw  $t1,-4($sp)
    sw  $t2,-8($sp)
    lw  $t0,_______($sp)
    lw  $t1,_______($sp)
    lw  $t2,_______($sp)
    add  $t0, $t0, $t1
    sra  $t1,_______,_______
    andi  $t0, $t0,1
    beq  $t0, $0,foo2_fin
    addi  $t1,_______,_______
foo2_fin:
    sw _______,0($t2)
    lw  $t0,_______($sp)
```

```
lw   $t1 ,_______( $sp )
lw   $t2 ,_______( $sp )
```

```
jr   $ra
```

Fonction	Taille Étiquette	Taille Index	Taille Déplacement
Blocs 16 octets			
Blocs 32 octets			

16 octets	Étiq	Index	Dép	Succès
0x2000				
0x2004				
0x2008				
0x2010				
0x2014				
0x2018				

Taux de succès : ——

32 octets	Étiq	Index	Dép	Succès
0x2000				
0x2004				
0x2008				
0x2010				
0x2014				
0x2018				

Taux de succès : ——

Inst / Cycle	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
lw $t0,0($a0)	LI	DI	EX	M	ER										
lw $t1,4($a0)															
and $t0,$t0,$t1															
sw $t0,8($a0)															
addi $a0,$a0,16															
addi $a1,$a1,-1															
bne $a1,$0,test															
lw $t0,0($a0)															

Diagramme avec envoi et branchement au plus tôt

```
# Programme réorganisé
 lw   $t0 ,0( $a0 )
_____________________
_____________________
```

```
_____________________
_____________________
bne   $a1 , $zero , test
_____________________
```

Instruction	Codage instruction	numéro *rs*	numéro *rt*	numéro ec
and $t0,$t0,$t1				
sw $t0,8($a0)				
bne $a1,$0,test				

Instruction	valeur ec	résultat UAL	valeur *cp*	valeur *cp* suivant
and $t0,$t0,$t1				
sw $t0,8($a0)				
bne $a1,$0,test				

Éléments de correction

toHexString(N3)	**0x**C134 0000
toHexString(N1& N2)	**0x**00CC CC00
toHexString(N1 | N2)	**0x**ABFF FFBA
toHexString(N1 ^ N2)	**0x**AB33 33BA
toHexString(N2 >> 1)	**0x**D5E6 F780
toHexString(N2>>> 1)	**0x**55E6 F780
toString(foo(4,7))	6
toString(foo2(4,7))	6
toHexString(foo(0x80000000,0x80000000))	0
toHexString(foo2(0x80000000,0x80000000))	0x80000000

foo2 calcule la $\frac{x+y}{2}$ comme un nombre non signé.

```
    addi  $a0, $0,4
    addi  $a1, $0,7
    jal  foo
    add  $a0, $0, $v0
    addi  $v0, $0,1
    syscall
    addi  $v0, $0,10
    syscall
foo:
#  $a0  et  $a1  paramètres
#  $v0  résultat
    sw  $t0,0($sp)
    add  $t0, $a0, $a1
    sra  $v0, $t0,1
    andi  $t0, $t0,1
    beq  $t0, $0,foo_fin
    addi  $v0, $v0,1
foo_fin:
    lw  $t0,0($sp)
    jr  $ra
```

```
  # passage par la pile
    sw  $t0,0($sp)
    addi  $t0, $0,4
    sw  $t0,-4($sp)
    addi  $t0, $0,7
    sw  $t0,-8($sp)
    addi  $t0, $0,0x2000
    sw  $t0,-12($sp)
    addi  $sp, $sp,-16
    jal  foo_pile
    addi  $sp, $sp,16
    lw  $a0,0x2000($0)
  # idem foo
foo_pile:
    sw  $t0,0($sp)
    sw  $t1,-4($sp)
    sw  $t2,-8($sp)
    lw  $t0,12($sp)
    lw  $t1,8($sp)
    lw  $t2,4($sp)
    add  $t0, $t0, $t1
    sra  $t1, $t0,1
    andi  $t0, $t0,1
    beq  $t0, $0,foo2_fin
    addi  $t1, $t1,1
foo2_fin:
    sw  $t1,0($t2)
    lw  $t0,0($sp)
```

```
lw   $t1,-4($sp)
lw   $t2,-8($sp)
```

```
jr   $ra
```

Calcule le et bit à bit des valeurs contenues dans **0x**2000 et **0x**2004 range le résultat **0x**2018 Calcule le et bit à bit des valeurs contenues dans **0x**2010 et **0x**2014 range le résultat **0x**2018

Fonction	Taille Étiquette	Taille Index	Taille Déplacement
Bloc 16	25	3	4
Bloc 32	25	2	5

Directe	Étiq	Index	Dép	Succès
0x2000	**0x**40	0	0	D
0x2004	**0x**40	0	4	S
0x2008	**0x**40	0	8	S
0x2010	**0x**40	1	0	D
0x2014	**0x**40	1	4	S
0x2018	**0x**40	1	4	S

Taux de succès : $\frac{4}{6}$

Assoc ens	Étiq	Index	Dép	Succès
0x2000	**0x**40	0	0	D
0x2004	**0x**40	0	4	S
0x2008	**0x**40	0	8	S
0x2010	**0x**40	0	10	S
0x2014	**0x**40	0	14	S
0x2018	**0x**40	0	18	S

Taux de succès : $\frac{5}{6}$

Inst / Cycle	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
lw $t0,0($a0)	LI	DI	EX	M	***ER*** *$t0*										
lw $t1,4($a0)		LI	DI	EX	M	ER									
and $t0,$t0,$t1			LI	DI	***EX*** *$t0*	***EX*** *$t1*	*M* *$t0*	ER							
sw $t0,08($a0)				LI	DI	DI	***EX*** *$t0*	M	ER						
addi $a0,$a0,16					LI	LI	DI	EX	M	ER					
addi $a1,$a1,-1							LI	DI	EX	*M* *$a1*	ER				
bne $a1,$0,test								LI	DI	*DI* *$a1*	EX	M	ER		
lw $t0,0($a0)											LI	DI	EX	M	ER

Diagramme avec envoi et branchement au plus tôt

```
# Programme réorganisé
lw   $t0,0($a0)
lw   $t1,4($t1)
addi  $a1, $a1,-1
and  $t0, $t0, $t1
sw   $t0,8($a0)
bne  $a1, $zero,testCache
addi  $a0, $a0,16
```

Instruction	Codage instruction	numéro *rs*	numéro *rt*	numéro ec
and $t0,$t0,$t1	**0x**0109 4024	8	9	8
sw $t0,8($a0)	**0x**AC88 0008	4	8	x
bne $a1,$0,test	**0x**14A0 FFF9	5	0	x

Instruction	valeur ec	résultat UAL	valeur *cp*	valeur *cp* suivant
and $t0,$t0,$t1	**0x**0	**0x**0	**0x**10	**0x**14
sw $t0,8($a0)	x	**0x**2008	**0x**14	**0x**18
bne $a1,$0,test	x	1	**0x**20	**0x**8

Entrées/sorties
- Erreur dépassement arithmétique
- Erreur instruction et adresses en lw et sw et adresse (voir cours