
ARCHITECTURE DES ORDINATEURS

Devoir Surveillé 1

1 heure 20 minutes
sans documents

- N.B.** : - Travaillez calmement. Ne bâclez pas vos réponses : il vaut mieux traiter correctement moins de questions que tout faire de travers.
- Les réponses aux questions doivent être argumentées et aussi concises que possible.
- Le barème est donné à titre indicatif.

Question 1 (5 points)

On considère des entiers codés sur 16 bits, interprétés comme des entiers relatifs codés en complément à deux. Chacun de ces entiers peut (et doit) s'écrire avec quatre chiffres hexadécimaux (on omettra le préfixe « 0x »). Soient les cinq entiers suivants :

M	N	P	Q	R
1418	F00D	DADA	3945	8080

- (1.1) (1 point)
Parmi ces 5 entiers, quels sont ceux qui sont négatifs ? Expliquez pourquoi.
- (1.2) (1 point)
Classez ces entiers du plus petit au plus grand (par exemple : « $M < N < P < Q < R$ », mais cette réponse choisie arbitrairement n'est probablement pas la bonne...). Justifiez la réponse sur votre copie.
- (1.3) (1 point)
Calculez $P + Q$. Pour cela, effectuez l'addition en binaire directement sur la copie — une réponse brute ne rapportera aucun point. Y a-t-il débordement (*overflow*) ?
- (1.4) (2 points)
Calculez $-N$ et $M - N$. Y a-t-il débordement (*overflow*) ? Comme pour la question précédente, le détail des calculs doit figurer sur la copie.

Question 2 (5 points)

On considère des nombres entiers signés à deux bits, codés en complément à deux, tel que le nombre X s'écrit par exemple $X = x_1x_0$. Le bit 1 est le bit de poids fort, et le bit 0 est le bit de poids faible.

- (2.1) (1 point)
À quelles valeurs entières signées les valeurs binaires 00, 01, 10 et 11 correspondent-elles ?
- (2.2) (2 points)
On veut créer un circuit calculant $N = n_1n_0$, l'opposé en complément à deux du nombre X qui lui est fourni. Donnez les formules de n_1 et n_0 à partir de x_1 et x_0 (vous pouvez vous aider en calculant la formule de la retenue c entre les deux colonnes). Dessinez le câblage de ce circuit avec des portes AND, OR et/ou XOR. Dans la suite, on pourra représenter ce circuit par une boîte NEG à deux entrées et deux sorties.
- (2.3) (2 points)
En utilisant éventuellement les circuits vus en cours, câblez un circuit ABS renvoyant la valeur absolue $A = a_1a_0$ du nombre $X = x_1x_0$ qui lui est fourni.

Question 3

(10 points)

Voici le texte d'un programme écrit en langage y86. Tous les commentaires ayant malencontreusement été effacés, il vous faut partir de zéro pour comprendre ce que fait ce programme, en répondant aux questions ci-dessous.

Attention : les réponses qui, au lieu d'expliquer, paraphrasent simplement le code (telles que « on soustrait %edx à %ebx ») ne rapporteront aucun point.

0x000:		.pos	0	(3.1)	(2 points)
0x000: 308400020000	main:	irmovl	pile,%esp		Quels sont les paramètres de la fonction <code>mystere</code> ? Expliquer quelle partie du code permet de répondre à cette question.
0x006: 308000010000		irmovl	m,%eax		
0x00c: a008		pushl	%eax		
0x00e: 308008010000		irmovl	t,%eax		
0x014: a008		pushl	%eax		
0x016: 8028000000		call	mystere	(3.2)	(1 point)
0x01b: c08408000000		iaddl	8,%esp		Quel est le rôle de l'instruction d'adresse 0x01b?
0x021: 400804010000		rrmovl	%eax,n		
0x027: 10		halt		(3.3)	(1 point)
0x028: a068		mystere: pushl	%esi		Aurait-on pu écrire de façon plus compacte en mémoire l'instruction d'adresse 0x030?
0x02a: 506408000000		mrmovl	8(%esp),%esi		
0x030: 308000000000		irmovl	0,%eax		
0x036: 2002		rrmovl	%eax,%edx	(3.4)	(4 points)
0x038: 501600000000	et1:	mrmovl	(%esi),%ecx		Que calcule la fonction <code>mystere</code> , et que renvoie-t-elle?
0x03e: 6211		andl	%ecx,%ecx		Pour cela :
0x040: 735d000000		je	et3		— Expliquez le rôle de chacun des registres utilisés au sein de cette fonction et leur évolution au cours du déroulement de la fonction.
0x045: 724c000000		jl	et2		— Expliquez le rôle des étiquettes <code>et1</code> , <code>et2</code> et <code>et3</code> .
0x04a: 6012		addl	%ecx,%edx		
0x04c: c08001000000	et2:	iaddl	1,%eax		
0x052: c08604000000		iaddl	4,%esi		
0x058: 7038000000		jmp	et1		
0x05d: 50640c000000	et3:	mrmovl	12(%esp),%esi		
0x063: 402600000000		rrmovl	%edx,(%esi)		
0x069: b068		popl	%esi	(3.5)	(1 point)
0x06b: 90		ret			Si ce code avait été produit par un compilateur C, quel aurait été le prototype de la fonction <code>mystere</code> ?
0x100:		.pos	0x100		
0x100: 00000000	m:	.long	0	(3.6)	(1 point)
0x104: 00000000	n:	.long	0		
0x108: 22013111	t:	.long	0x11310122		Quelle est la valeur contenue à l'adresse 0x0100 à la fin du programme?
0x10c: b74223da		.long	0xDA2342B7		
0x110: 20411131		.long	0x31114120		
0x114: 9180bdfa		.long	0xFABD8091		
0x118: 00000000		.long	0x00000000		
0x200:		.pos	0x200		
0x200: 00000000	pile:	.long	0		