|  |  |
| --- | --- |
| Ordinateurs: structure et applications | Nom/Matricule : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| GIF-1001, Hiver 2015 |  |

## Travail Pratique 1

## LES BUS

# Ce travail pratique vaut 4% de la note totale du cours. À faire individuellement, il contient 24 questions obligatoires comptabilisées sur un total de 100 points, ainsi que 2 questions bonus pour 10 points supplémentaires. Il est donc possible d’obtenir une note de 110%. Le laboratoire est à remettre au plus tard mardi 3 février à 23h59. La procédure de retard détaillée dans le plan de cours s’applique : 3 jours de retard « gratuits » pour toute la session, 10% par jour de retard lorsque les jours gratuits sont utilisés, et note = 0 au-delà de 30% de pénalité.

## OBJECTIFS

# Ce travail pratique vise les objectifs suivants:

# Comprendre le fonctionnement des bus d’un micro-ordinateur

# Comprendre l’adressage de la mémoire et des entrées/sorties

# Comprendre la lecture, le décodage et l’exécution des instructions par un microprocesseur

## PRÉPARATION

# Dans ce travail pratique, nous utiliserons un simulateur d'ordinateur très rudimentaire. Il est tellement rudimentaire qu’il ne possède même pas de microprocesseur : VOUS serez le microprocesseur. Vous contrôlerez les bus d’adresse, de données et de contrôle via des boutons virtuels.

# Vous devrez lire, décoder et exécuter des instructions dans une mémoire d’instructions non-volatile (« Read-Only Memory », ou ROM). Ces instructions vous demanderont de manipuler des variables dans la mémoire de données (« Random Access Memory », ou RAM) et d’accéder aux périphériques. L’ordinateur est doté d’un périphérique d’entrée (un clavier ASCII) et d’un périphérique de sortie (un écran à 7 segments).

# Vous utiliserez un simulateur d’ordinateur disponible sur le site web du cours, où le microprocesseur est manquant. Vous jouerez le rôle du microprocesseur, et répondrez à des questions concernant le fonctionnement de l’ordinateur.

# La section 8 du guide de l’utilisateur décrit le jeu d’instructions supporté par l’ordinateur, et vous sera utile pour répondre à plusieurs des questions.

## QUESTIONS

# 1. (4 points) Quelles sont les interfaces (boutons) par lequel le microprocesseur (donc vous) communique avec le reste du circuit?

# 2. (4 points) Où sont les registres R0, R1, R2, R3 et PC dans l’ordinateur?

# 3. (2 points) Quel est le plus grand nombre non-signé pouvant être stocké d’un coup par ce système? Le plus petit?

# 4. (2 points) Quel est le plus grand nombre signé (en représentation complément 2) pouvant être stocké d’un coup par ce système? Le plus petit?

# 5. (4 points) Combien d’adresses différentes le microprocesseur peut-il générer?

# 6. (4 points) Quelles valeurs des bus de contrôle et d’adresse permettent au microprocesseur (vous) d’envoyer des données sur le bus de données?

# 7. (2 points) Combien d’adresses de mémoire d’instructions (ROM) utilisons-nous?

# 8. (2 points) Combien d’adresses de mémoire de données (RAM) utilisons-nous?

# 9. (4 points) On désire lire une valeur contenue en mémoire de données (RAM). Décrivez le mécanisme qui permet au microprocesseur de différentier entre la mémoire de données (RAM), la mémoire d’instructions (ROM), et les périphériques lorsque le bus de contrôle est en mode « lecture ».

# 10. (4 points) Pour notre périphérique de sortie (écran), on a ici un « décodage partiel », ce qui est plus économique mais n’assure pas une adresse unique à notre périphérique. Donnez la liste de toutes les adresses, en binaire, qui peuvent commander notre périphérique de sortie. Vous pouvez indiquer les adresses par intervalles, au lieu de toutes les lister indépendamment.

# 11. (10 points) Déterminez la carte mémoire (« memory-map ») de ce système en spécifiant quelle partie de l’ordinateur (RAM, ROM, périphérique d'entrée, périphérique de sortie, nul) répond à chaque adresse :

|  |  |
| --- | --- |
| Adresse | Circuit |
| 0x00 à 0x3F |  |
| 0x40 à 0x7F |  |
| 0x80 à 0xBF |  |
| 0xC0 à 0xFF |  |

# 12. (4 points) Si on ajoutait des lignes sur le bus de données, devrait-on aussi ajouter des lignes sur le bus d’adresse? Pourquoi?

# 13. (8 points) Un utilisateur écrit la chaine de caractères suivante au clavier : « AB ». Répondez aux deux questions suivantes.

# Écrivez un programme (2 instructions devraient suffire) qui permet au microprocesseur de mettre la valeur du clavier dans le registre R0 ?

# Quelle est la valeur contenue dans le registre R0 après avoir exécuté ces instructions ? Pourquoi ? Indice : pensez à la convention utilisée par le clavier pour représenter les chaines de caractères.

# 14. (4 points) Décodez l’instruction suivante en écrivant le code assembleur correspondant : « 0x41AB ». Indice : la section 8 du guide de l’utilisateur vous sera utile.

# 15. (4 points) Décodez l’instruction suivante en écrivant le code assembleur correspondant : « 0x6101 ».

# 16. (2 points) Comment le microprocesseur pourrait-il produire une collision sur le bus de données?

# 17. (2 points) Est-il possible que deux composantes puissent écrire des données en même temps sur le bus de données? Pourquoi?

# 18. (4 points) Lorsque le signal "lecture" est actionné, est-ce le processeur, la mémoire ou les deux qui lisent le bus de donnée?

# 19. (4 points) Le microprocesseur sait-il si les instructions qu’il exécute sont en ROM, en RAM ou proviennent d’un périphérique?

# 20. (4 points) La valeur 0x1234 est placée dans le registre R2. Cette même valeur est aussi écrite en mémoire, à une adresse stockée dans le registre R3. Écrivez deux instructions assembleur différentes permettant de placer la valeur 0x1234 dans le registre R0, et qui utilisent R2 et R3.

# 21. (4 points) Écrivez les deux instructions de la question précédente en hexadécimal (sur 16 bits).

# 22. (10 points) Chargez le programme contenu dans le fichier « tp1.asm » dans la mémoire d’instructions. Chargez aussi le fichier « tp1.xml » dans la mémoire de données. Exécutez le programme en jouant le rôle du micro-processeur, et répondez ensuite aux deux questions suivantes (n’oubliez pas de réinitialiser la simulation avant de procéder) :

# Expliquez dans vos mots ce que fait le programme dans le fichier « tp1.asm »?

# Quel est le contenu final des registres?

# 23. (4 points) Pour le microprocesseur, y-a-t-il une différence entre l’accès à la mémoire RAM et l’accès à un périphérique?

# 24. (4 points) Quel est le nombre maximal de registres pouvant être supportés par le jeu d’instructions en section 8 du guide de l’utilisateur?

# 25. (5 points bonus) Écrivez un programme qui copie sa première instruction à l’adresse 0x3F de la mémoire de données (RAM). Votre programme devra débuter à la première case mémoire de la mémoire d’instructions (ROM). 4 instructions devraient vous suffire.

# 26. (5 points bonus) Après l’exécution de votre programme ci-haut, quelle sera la valeur à l’adresse 0x3F dans la mémoire de données (RAM)?