Département de génie électrique et de génie informatique
Université de Sherbrooke

ÉVALUATION FORMATIVE

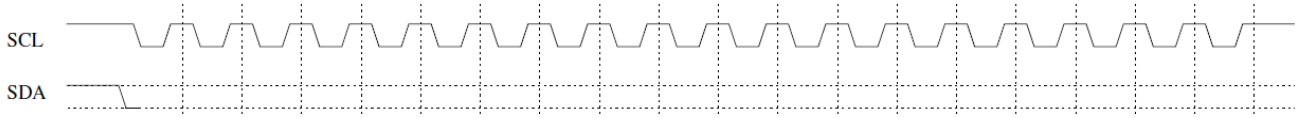
Exercice 1 Protocole I2C

On souhaite envoyer un octet à un esclave via le protocole I2C. Les paramètres sont les suivants:

- Adresse de l'esclave: 0x6F
- Données à transmettre: 0xFA

(a) Quelle est la séquence précise d'opérations à réaliser pour effectuer ce transfert selon le protocole I2C?

(b) Complétez le chronogramme ci-dessous pour illustrer le transfert complet, en indiquant par annotation les évènements importants.



(c) Dessinez un chronogramme I2C de lecture d'un registre identifié par B7:B0 (MSB:LSB) d'un périphérique esclave à l'adresse A6:A0, et qui contient la valeur D7:D0. Utilisez les symboles S (start), P (stop), A (ack) et NA (no-ack) pour les bits de contrôle. Indiquez clairement (par une couleur ou autre) qui contrôle (maître ou esclave) le signal SDA.

Exercice 2 Écriture de code assembleur

Considérez le code C suivant :

```
1 int valeur = 64;
2
3 int main() {
4       int n = 0;
5       int s = 0;
6
7       while ( s <= valeur) {
8       s = s + 2 * n + 1;          /* mise à jour de la somme */
9       n = n + 1;                  /* incrémentation de l'index */
10       }
11 }
```

(a) Comment feriez-vous en assembleur pour calculer $2 \cdot n + 1$ sans avoir recours à une multiplication ni une addition?

(b) Écrivez un programme en langage assembleur MIPS qui réalise la même fonction, de manière à ce que la valeur finale de n se trouve dans $t0 en fin d'exécution.

(c) Écrivez un programme qui change l'état des bits 15, 16 et 17 du mot de 32 bits à l'adresse 0x10010000.

Exercice 3 Analyse de code assembleur

Soit le code assembleur suivant :

```
1 .data 0x10010000
2 var:   .word 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
3
4 .text
5
6 main:
7       la $t1, var
8       lw $t3, 4($t1)
9       lw $t4, 16($t1)
10      addu $t5, $t4, $t3
11
12      .end main
```

(a) Donnez la valeur du registre de destination après les instructions aux lignes 7, 8 et 9.

(b) Commenter le code suivant.

(c) Que contient $t5 en fin d'exécution?

Exercice 4 Architecture MIPS

(a) Donner les modes d'adressage des instructions MIPS32.

(b) Expliquez la différence entre une instruction, une pseudo-instruction et un mode d'adressage non-supporté.

(c) Donnez des exemples de mode d'adressage non compatible nativement avec le MIPS32.

(d) Comment fait-on en langage assembleur MIPS pour transférer une mot de 32 bits de l'adresse 0x10010000 à l'adresse 0x100100F8?

Exercice 5 Microcontrôleurs

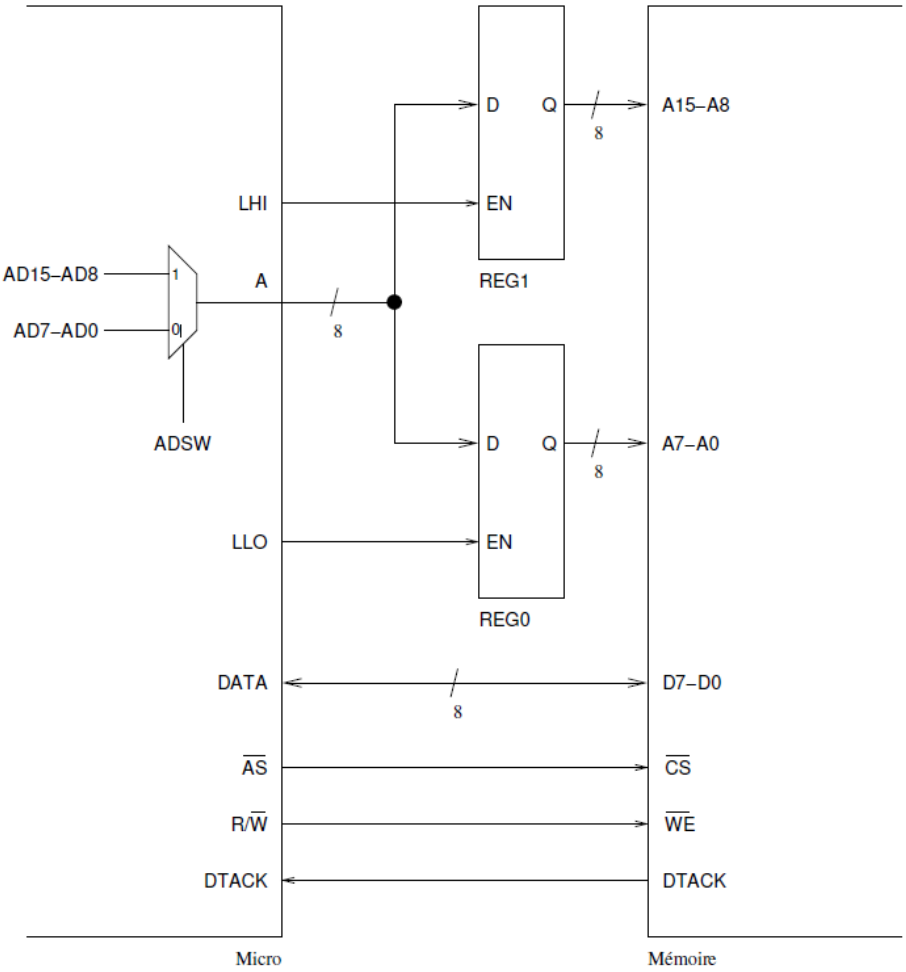
(a) En quoi consiste la protection de contexte en programmation mixte C/assembleur?

(b) Décrivez la convention d'appel d'une fonction assembleur en ce qui concerne les paramètres d'entrée et de retour.

(c) Qu'est-ce qui justifie l'utilisation du mécanisme d'interruption ou d'interrogation lorsque le microcontrôleur doit recevoir des données par un port externe? Lequel choisir et pourquoi?

Exercice 6 Séquence d'évènements dans les interfaces

Vous disposez d'un microcontrôleur minimaliste n'ayant que 8 lignes d'adresse que vous souhaitez interfacer avec une mémoire de 64K. Un circuit d'interface (illustré ci-dessous) a été conçu à cette fin incluant 2 registres de 8 bits (REG0 et REG1), ce qui permet de capturer l'adresse complète (16 bits) en deux temps et de déclencher l'opération de lecture ou d'écriture par la suite. On a une adresse complète AD15-AD0 à l'intérieur du microcontrôleur et un multiplexeur interne permet de sélectionner la moitié haute ou basse de celle-ci vers les lignes d'adresses externes A. Les lignes LH1 et LH0 sont raccordées aux entrées "enable" des registres REG1 et REG0, respectivement, et permettent d'activer la capture des données présente à leur entrée D. Le contenu d'un registre est toujours présent à sa sortie Q. Une fois l'adresse complète présente sur les lignes d'adresse de la puce mémoire, la ligne AS (Adress Strobe) passe à '0' et active ainsi l'entrée CS (Chip Select) de la puce mémoire. La puce mémoire va alors utiliser l'adresse présentée et les données (s'il s'agit d'une écriture) pour effectuer l'opération souhaitée (selon la valeur de la ligne R=W qui vaut '1' pour une lecture et '0' pour une écriture). Le microcontrôleur doit ensuite attendre que la ligne DTACK monte à '1' pour signifier que l'accès mémoire est complété.

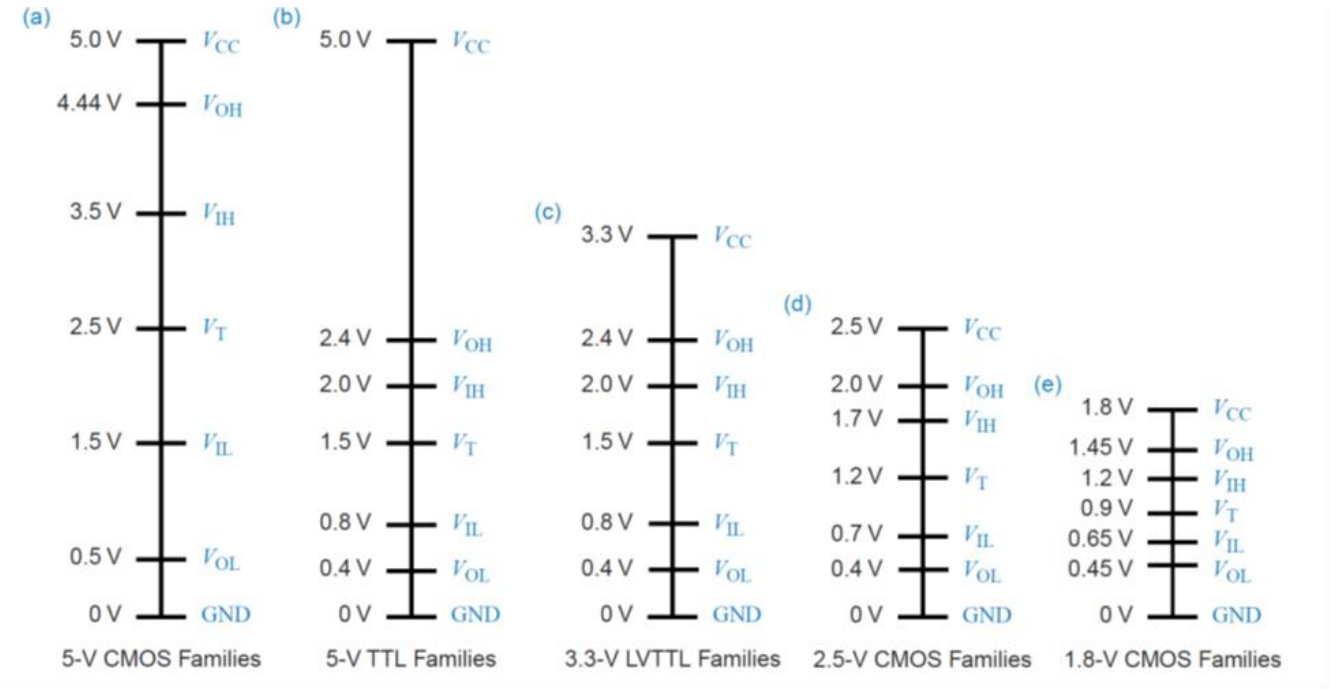


(a) Complétez le graphe ci-dessous de la machine à états qui génère les signaux de commande et qui effectue en continu des écritures en mémoire. Vous n'avez pas à gérer le bus de données.

(b) Est-il vraiment nécessaire d'avoir deux registres pour la gestion de l'adresse? Expliquez.

Exercice 7 Compatibilité électrique

Si un micro-processeur doit lire les données d'un capteur de température utilisant un protocole de communication I2C dont les niveaux de tensions sont les suivants 0,2 V et 3,4 V. Selon la figure ci-dessous, quelles familles sont compatibles avec ce capteurs?



Exercice 8 Code en C

Expliquer ce que font le lignes suivantes en C, vous avez doit à la librairie fournie pour l'APP.

(a) unsigned char val1 = PMODS_GetValue(0,1);

(b) unsigned int val2 = ADC_AnalogRead(2);

(c) LCD_WriteStringAtPos(msg, 0, 0);

Exercice 9 Configuration des ports

Donner les registres pour configurer et utiliser les ports.