

Exercices pour le cours de GIF-3002, Systèmes Microprocesseurs et Interfaces, Partie 1 de 2

Ce document contient des exercices accompagnant les notes de cours de GIF-3002 afin de consolider et approfondir certaines notions vues en classe.

Les exercices sont divisés par cours ou par sujet présenté par l'enseignant.

De nombreuses personnes ont contribué à la réalisation de ces exercices, souvent au détriment de leur sommeil. Plusieurs exercices sont le produit d'un effort additionnel, à une heure indue. Veuillez nous pardonner les fautes, les omissions ou la mise en page minimaliste...

Bon apprentissage,

Etienne Tremblay, Chargé de cours
Marc-André Béland, Assistant de cours
Patrick Clément-Bonhomme, Assistant de cours
Marc-André Gardner, Assistant de cours
Bernard Lebel, Assistant de cours

Cours 1 : Introduction

Q1.1 *Donnez l'ordre d'apparition des technologies suivantes à l'échelle mondiale :*

- 1. Les transistors à effet de champs (CMOS)*
- 2. Les circuits intégrés*
- 3. Les transistors bipolaires (BJT)*
- 4. Les ordinateurs*
- 5. Les microprocesseurs*

Q1.2 *Le processeur Intel Core i7, produit en 2010, contient environ 1.17 milliards de transistors. Combien de transistors les processeurs de 2015 devraient-ils contenir si les dimensions du processeur ne changent pas, si on utilise toujours la technologie la plus récente et si la loi de Moore est respectée?*

Q1.3 *Dans le domaine des microprocesseurs pour PC, nommez les deux compagnies ayant le plus grand volume de ventes mondialement. Dans le domaine des microcontrôleurs, nommez trois des cinq compagnies ayant le plus grand volume de ventes mondialement. Dans le domaine des architectures de microcontrôleurs, nommez trois des cinq architectures les plus communes mondialement.*

Q1.4 *Dans un système microprocesseur, qui impose les données sur le bus de données, qui impose les adresses sur le bus d'adresses et qui gère les lignes de contrôle?*

Q1.5 *Afin de minimiser le nombre de broches du microprocesseur, de minimiser le nombre de fils dans un système microprocesseur et de minimiser le nombre de contrôleurs de bus nécessaires dans le microcontrôleur, il y avait traditionnellement un seul bus, comprenant des lignes d'adresses, de données et de contrôle, partant du microprocesseur. Pourquoi les microprocesseurs modernes ont-ils souvent plusieurs bus?*

Q1.6 *Vous écrivez un programme en C ayant la ligne suivante : $\langle a = b + c \rangle$. Sachant que :*

- a, b, et c sont des variables de type "short" (16 bits) aux adresses 0x1000, 0x1002 et 0x1004*
- Le microprocesseur exécutant le programme a 4 registres 16 bits généraux, R0, R1, R2 et R3.*
- Le microprocesseur exécutant le programme a une architecture LOAD/STORE (LOAD Reg, [adresse]) et supporte l'instruction ADD destination, source 1, source2.*

Écrivez la séquence d'instructions à être exécutée par le programme qui pourrait correspondre à la ligne de code $\langle a = b + c \rangle$.

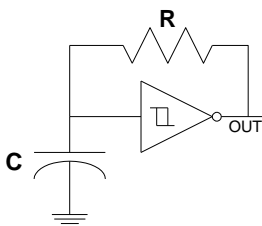
Q1.7 *Dites quelles instructions se retrouvent dans le microprocesseur et décrivez le(s) programme(s) auxquels ces instructions appartiennent. Dites également quelles composantes servent à contenir les données dans un microprocesseur.*

Q1.8 Pour les applications suivantes, dites s'il faudrait utiliser un microprocesseur ou un microcontrôleur et dites pourquoi :

1. Contrôleur de grille-pain
2. Contrôleur d'automobile
3. Téléphone cellulaire
4. Ordinateur portable
5. Supercalculateur

Q1.9 Décrivez deux méthodes utilisées pour programmer la FLASH d'un microcontrôleur, sachant que l'adresse initiale, au démarrage, du compteur de programme des microcontrôleurs (PC) désigne souvent la FLASH ou une autre mémoire non-volatile.

Q1.10 Quel signal est généré sur la broche OUT du circuit suivant?



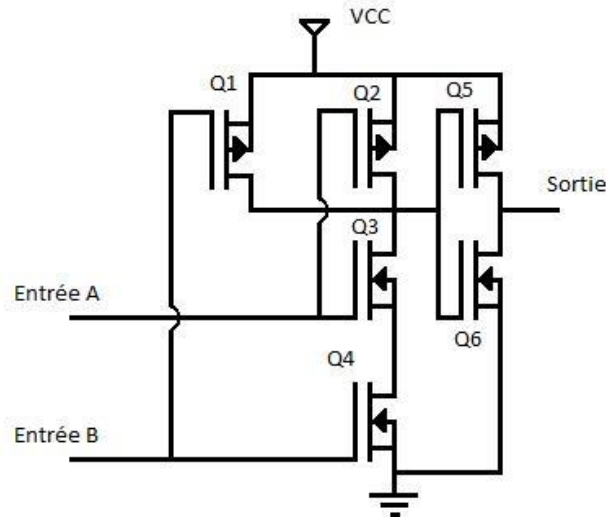
Q1.11 Décrivez les tâches d'un microprocesseur.

Q1.12 Pour quelle raison la logique trois états existe-t-elle?

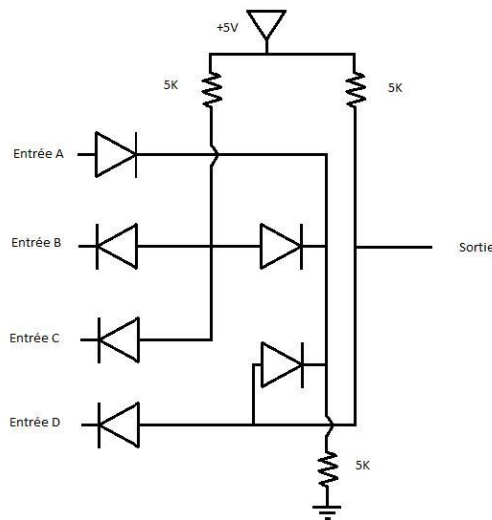
Q1.13 Quelle est la différence entre un microprocesseur et un microcontrôleur? Donnez les avantages et inconvénients de chacun.

Q1.14 Il faut brancher le bus de données d'un microcontrôleur à deux mémoires pour un interfaçage en lecture et écriture. Chaque ligne du microcontrôleur peut fournir jusqu'à 10 mA. Or, chaque entrée sur les mémoires nécessite 6 mA. Comment est-il possible de régler ce problème?

Q1.15 Dans le circuit suivant, les entrées peuvent être égales soit au VCC, soit à la masse. Identifier le type de chaque MOSFET du circuit. Écrivez la table de vérité de ce circuit. Quelle est la fonction logique de ce circuit?



Q1.16 Quelle est la fonction logique du circuit suivant? (On assume que la perte de tension dans chaque diode est de 0,7V et que les quatre entrées peuvent être seulement 5V ou 0V, c'est-à-dire un '1' ou un '0' logique)

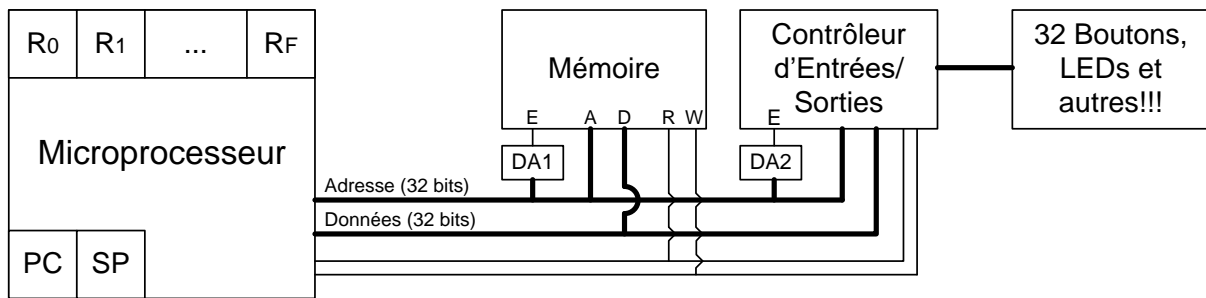


Q1.17 Pourquoi une architecture Harvard est-elle très avantageuse lorsque le microprocesseur a un pipeline d'instructions?

Q1.18: Vous avez un système microprocesseur avec deux mémoires, une mémoire FLASH de 128K-mots et une mémoire RAM de 64K-mots. Dessinez le bus d'adresse de ce système avec le décodeur d'adresse. Notez que $K = 2^{10}$ et que vous devez déterminer les éléments non-établis du problème (exemple: l'adresse de base des mémoires) comme il vous plaira.

Q1.19: Les notes de cours décrivent sommairement le protocole NMEA 0183 pour représenter les fractions. Dites pourquoi ce protocole est peu utilisé dans l'industrie pour représenter les autres fractions.

Q1.20 (récapitulatif) Supposons le système suivant :



Dans ce système :

- **Le microprocesseur fonctionne sur 32 bits : les registres sont 32 bits, les adresses sont 32 bits, le bus de données a 32 bits et toutes les instructions sont sur 32 bits.**
- **Le microprocesseur a 16 registres tout usage utilisés pour faire des calculs.**
- **Le microprocesseur a un registre PC, Program Counter, indiquant l'adresse de la prochaine instruction à exécuter. Le PC est automatiquement incrémenté de 4 après la lecture d'une instruction en mémoire.**
- **Le microprocesseur a un registre SP, le Stack Pointer. Lors d'un PUSH, la donnée est mise sur la pile à l'adresse indiquée par SP, puis SP est automatiquement décrémenté de 4. Lors d'un POP, SP est incrémentée de 4, puis la donnée de l'espace mémoire indiqué par SP est stockée à l'endroit requis par l'instruction.**
- **Lorsqu'une interruption se produit, le microprocesseur exécute automatiquement les tâches suivantes, dans l'ordre :**
 - o **Mettre la valeur de PC actuelle sur la pile (SP est automatiquement décrémenté après l'opération)**
 - o **Lire l'adresse 4*(numéro d'interruption).**
 - o **Prendre la valeur lue et la mettre dans PC : la table des vecteurs d'interruption commence à l'adresse 0...**
- **Le microprocesseur supporte les instructions décrites ci-dessous (et plus!). L'opcode des instructions est sur 8 bits et les paramètres de chaque instruction sont sur 24 bits. Il faut 4 bits pour indiquer un des 16 registres tout-usage du microprocesseur.**

Opcode	Paramètres
bits 31 à 24	bits 23 à 0

Tableau 1. Instruction sur 32 bits.

- **La mémoire est activée lorsque l'adresse, sur le bus d'adresse, est entre 0x00000000 et 0x0004FFFF : le décodeur d'adresse DA1 active la mémoire (Enable) lorsque l'adresse est entre ces valeurs.**
- **Un contrôleur d'entrées sorties est activé lorsque les adresses sont 0x00060000 et 0x00060004. Ce contrôleur permet de lire ou d'écrire des entrées sorties digitales en fonction du registre de direction (0 = entrée, 1 = sorties). Le registre de valeur indique la valeur à mettre sur les sorties ou la valeur lue des entrées.**

Répondez aux questions en fonction des tables suivantes :

Mémoire		Instruction	Opcode	Registre	Adresse
Adresse	Valeur	LOAD Rx, [Mem]	0x10	4 bits	20 bits
0x00048100	0xFFFFFFFF	Store Rx, [Mem]	0x11	4 bits	20 bits
0x000480FC	0xFFFFFFFF				
0x000480F8	0xFFFFFFFF		Opcode	Registre	
0x000480F4	0xFFFFFFFF	PUSH Rx	0x20	4 bits	
...	...	POP Rx	0x21	4 bits	
0x00040008	0x00000003				
0x00040004	0x00000002		Opcode	Rdest	Rsrc1, Rsrc2
0x00040000	0x00000001	ADD Rdest, Rsrc1, Rsrc2	0x30	4 bits	8 bits
...	...				
0x00000220	0x40FFFFFF		Opcode		
0x0000021C	0x21500000	Retour d'interruption	0x40		
0x00000218	0x21600000				
0x00000214	0x117480FC	Périphérique contrôlant 32 broches			
0x00000210	0x30765FFF	Registre	Adresse	Valeur initiale	
0x0000020C	0x10660004	Direction	0x00060000	0x00000000	
0x00000208	0x10540008	Valeur	0x00060004	0x00000000	
0x00000204	0x20600000				
0x00000200	0x20500000	Valeur initiale des registres			
...	...	R4	0x00000004		
0x00000008	0x00000400	R5	0x00000005		
0x00000004	0x00000200	R6	0x00000006		
0x00000000	0x00000100	R7	0x00000007		

a) Si PC vaut 0x00000400 et SP vaut 0x000481000 lorsqu'une interruption #1 se produit, quelles seront les valeurs de PC et SP lorsque la première instruction de l'ISR (la routine traitant l'interruption) sera lue? Quelle valeur de la mémoire changera?

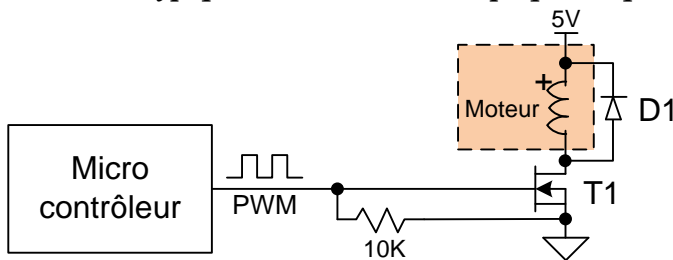
b) Quelle est la première instruction exécutée dans l'ISR? Pourquoi retrouve-t-on habituellement cette instruction au début d'une ISR?

c) Décrivez les signaux qui apparaîtront sur les bus du système lorsque la troisième instruction de l'ISR sera lue, puis exécutée.

d) Quelles seront les valeurs des registres R5, R6 et R7 après l'addition?

e) Quelles valeurs retrouvera-t-on aux adresses de 0x000480F4 à 0x000480100 après l'exécution de l'ISR?

Q1.21: On utilise presque toujours une diode en parallèle avec une inductance (ou un moteur) lorsque cette inductance est reliée à un interrupteur (habituellement un transistor; dans un power supply SMPS par exemple). La figure ci-dessous illustre ce propos avec un contrôleur typique de moteur DC. Expliquez à quoi sert cette diode (D1 sur la figure)?



Q1.22 : Un programmeur écrit la ligne de code suivante en C: `Toto = 3.2;`. Toto est défini comme un float. Le code est compilé sur un microprocesseur ARM Cortex M4F ayant un FPU. Ce microprocesseur a 32 registres 32 bits S0 à S31 disponibles pour manipuler les fractions et tout un jeu d'instruction réservé à cet effet. Toutes les instructions utilisées pour manipuler des fractions commencent par V. Par exemple, on retrouvera VLDR pour lire une variable en mémoire et mettre sa valeur dans un registre de S0 à S31. Lorsque le programmeur regarde le code assembleur correspondant à `Toto = 3.2;`, il retrouve les instructions suivantes :

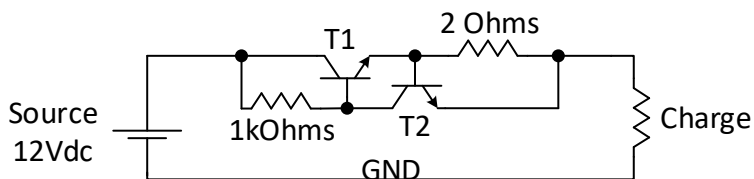
```

;Code pour Toto = 3.2;
LDR R0, [PC - 24]
LDR R1, [PC - 24]
STR R0, [R1]

```

Expliquez pourquoi ce code ne contient aucun registre relié aux fractions, aucune instruction reliée aux fractions et pas de référence explicite à Toto. En d'autres mots, expliquez comment le code fonctionne.

Q1.23 : On retrouve le circuit suivant pour limiter le courant provenant de la source, lorsque la charge est trop grande (résistance trop faible ou court-circuit). Expliquez comment ce circuit fonctionne :



Cours 2 Micro-Matériel

Q2.1 Énumérez 8 critères qui devraient guider la sélection d'un microprocesseur.

Q2.2 Dites quels mots sont définis ci-dessous :

<i>Définition</i>	<i>Mot</i>
<i>Circuit ayant pour fonction de faire un reset du microprocesseur lorsque celui-ci cesse de fonctionner normalement</i>	
<i>Ensemble de programmes servant à évaluer les performances d'un microprocesseur</i>	
<i>Se dit d'un contrôleur d'interruptions lorsque les adresses des routines d'interruptions sont configurables et lues dans une table d'adresses.</i>	
<i>Circuit permettant de transférer des données de la mémoire aux périphériques sans intervention du CPU</i>	
<i>Circuit générant des ondes rectangulaires à fréquence variable et à largeur variable</i>	

Q2.3 Lorsqu'on conçoit un circuit électronique avec un microprocesseur, quels sont les circuits de base accompagnant le microprocesseur qui doivent être élaborés? (Exemple de circuit de base : les alimentations du microprocesseur...).

Q2.4 Pour quelle(s) raison(s) les microcontrôleurs opèrent-ils habituellement entre 10MHz et 100MHz alors que les microprocesseurs opèrent à des vitesses beaucoup plus grandes?

Q2.5 Qu'est-ce qu'une Nested Vectored Interruption?

Q2.6 Pourquoi est-il déconseillé de faire une horloge indiquant le temps à l'aide de l'oscillateur interne d'un microcontrôleur? Comment peut-on faire une telle horloge avec le kit microcontrôleur du cours?

Q2.7 Quel est l'intervalle entre les différentes tensions pour un ADC 12 bits dont la tension d'entrée varie entre -3,3 V et 3,3 V ?

Q2.8 Pourquoi les microprocesseurs ont-ils besoin de plusieurs broches d'alimentation?

Q2.9 Dites pourquoi certaines broches de microcontrôleur ont parfois le rôle d'adresse et de données, multiplexé temporellement.

Q2.10 Qu'est-ce qu'un MAX232?

Q2.11 Dans un microcontrôleur moderne, quelle composante limite habituellement la vitesse de lecture et d'exécution des instructions? Quelles stratégies sont utilisées pour contourner les limites de cette composante?

Q2.12 Quelles stratégies peuvent être utilisées pour réduire la consommation de puissance d'un microcontrôleur?

Q2.13 Dans un contexte de microcontrôleur, que signifient les acronymes BGA, QFP, DIP?

Q2.14 Pourquoi l'identifiant unique d'un microcontrôleur n'est-il pas habituellement écrit dans la FLASH du microcontrôleur?

Q2.15 Pourquoi permet-on de commencer l'exécution de code après un reset à plusieurs adresses, habituellement selon certaines broches du microcontrôleur?

Cours 3 Micro-Logiciel

Q3.1 Vous retrouvez les instructions et données suivantes dans la mémoire d'un microcontrôleur. Assumez qu'un reset vient de se produire et que l'adresse du vecteur d'interruption traitant le reset est 0x00000004 et répondez aux questions suivantes (le registre R15 est le Program Counter).

Note 1: Le code qui suit peut contenir des erreurs sur le -8 et le -6 des LDR R1, [R15 - 8] et LDR R1, [R15 - 6] en raison du pipeline d'instruction du microcontrôleur. Afin de répondre aux questions, considérez que le PC (R15) ne varie pas pendant l'exécution d'une instruction.

Note 2: Assumez que le microprocesseur opère en little endian. Si vous retrouvez 0x12345678 à l'adresse 0x00001000, cela signifie qu'il y a 0x78 à l'adresse 0x00001000, 0x56 à l'adresse 0x00001001, 0x34 à l'adresse 0x00001002 et 0x12 à l'adresse 0x00001003.

Adresse	Valeur
0x00000000	0x00000000
0x00000004	0x00001004
...	
0x00001000	0x12345678
0x00001004	0x20000000
0x00001008	LDR R1, [R15 - 8]
0x0000100A	LDR R2, [R15 - 6]
0x0000100C	LDR R3, [R2+2]
0x0000100E	ADD R1, R2, R3
0x00001010	ADD R1, R1, 0x1234
...	
0x20000000	0xAAAA1111
0x20000004	0x2222BBBB

1. On utilise six octets de mémoire pour mettre 0x12345678 dans le registre R1 du micro processeur. Peut-on optimiser le code pour faire la même opération avec moins d'octets? Si oui, comment? Si non, pourquoi?
2. À quelle adresse devrait se retrouver l'instruction suivant cette séquence d'instruction?
3. Donnez la valeur du registre modifié par chaque instruction de la séquence. Dites quelle sera la valeur de R1 après le premier LDR, la valeur de R2 après le second LDR et ainsi de suite.

Q3.2 Qu'est qu'une banque de registre ou une banque de mémoire? Pourquoi utilise-t-on des banques de registres ou de mémoire?

Q3.3 Qu'est-ce qu'une architecture Load/Store?

Q3.4 À quoi sert le bit-banding du ARM_Cortex-M4? Quelle séquence d'instruction est optimisée par le bit-banding? Donnez un exemple de code assembleur.

Q3.5 Quelle séquence d'opérations effectue le microprocesseur ARM Cortex M4 lorsqu'il entre ou sort d'une interruption?

Q3.6 Nommer les différents modes d'adressage du ARM Cortex M4?

Q3.7 Quelle(s) composante(s) matérielle(s) place(nt) les mémoires et les périphériques aux adresses indiquées dans le « memory map »?

Q3.8 Dites si les énoncés suivants sont vrais ou faux. S'ils sont faux, dites pourquoi.

#	Énoncé	V/F
A	Dans un système PMIO, il y a une adresse 0 pour les périphériques et une adresse 0 pour les mémoires. Dans les systèmes MMIO, il y a une seule adresse 0.	
B	Toutes les mémoires ont un registre qui détermine la première adresse de la mémoire dans le système.	
C	Dans tous les systèmes MMIO, les instructions qui accèdent à la mémoire accèdent aussi aux périphériques. Dans tous les systèmes PMIO, les instructions qui accèdent aux périphériques activent des signaux de contrôle qui permettent de décoder une adresse de périphérique.	
D	Il faut que chaque adresse du système désigne un mot de mémoire ou un périphérique pour que le système fonctionne.	

Q3.9 Qu'est qu'un alias d'adresses?

Q3.10 Nommez et décrivez deux stratégies utilisées dans le cœur ARM pour diminuer le temps de latence moyen avant d'entrer dans une interruption.

Q3.11 Pourquoi la table des vecteurs d'interruptions du cœur ARM commence-t-elle par le vecteur de reset et par la valeur initiale du stack pointeur?

Q3.12 [Enrichissement] Le ARM Cortex M4 contient une banque de registres pour le pointeur de pile. Cela permet de sauver beaucoup de mémoire RAM lorsqu'un système d'exploitation préemptif est utilisé. Expliquez comment et pourquoi.

Cours 4 Mémoires, Matériel

Q4.1 Énumérez les signaux nécessaires afin de lire une mémoire ROM et indiquez le rôle de chacun des signaux.

Q4.2 Expliquez le fonctionnement de la mémoire SRAM.

Q4.3 Expliquez le fonctionnement de la mémoire DRAM.

Q4.4 Expliquez pourquoi les mémoires DRAM ont, pour une même quantité de broches, plus de capacité que les mémoires SRAM qui ne font pas multiplexage adresses/données.

Q4.5 Les chronographes suivants sont-ils des chronographes de lecture ou d'écriture de la mémoire? Expliquez pourquoi?

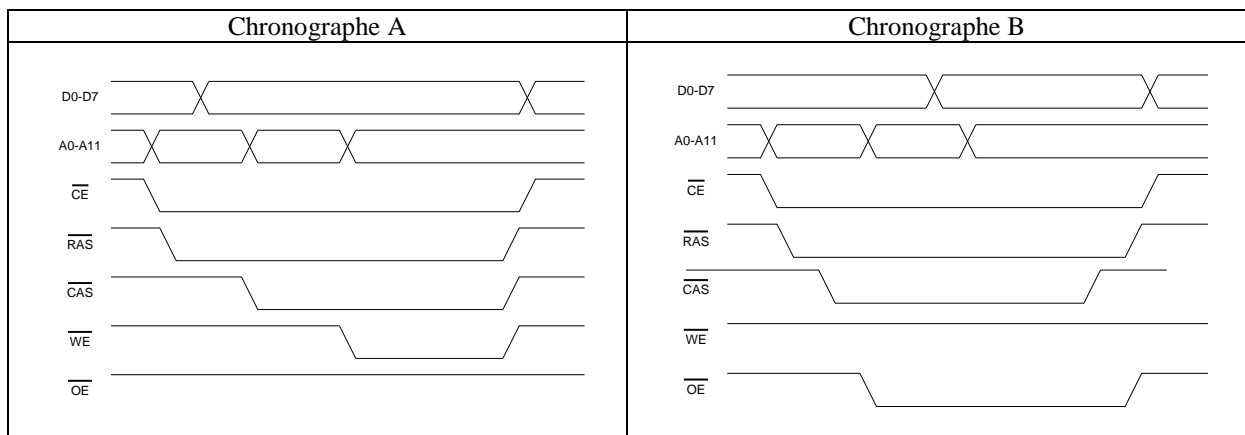


Figure 1 - Accès à la mémoire DRAM

Q4.6 Parmi ces mémoires non-volatiles (FLASH, EEPROM et FRAM), indiquez celle que vous utiliseriez dans les applications suivantes (la meilleure mémoire est celle qui coûte le moins cher, vous avez une quantité illimitée de mémoire non-volatile!). Indiquez pourquoi.

1. Vous mesurez la température à toutes les 100ms et vous devez sauvegarder toutes les températures lues dans le dernier 48 heures dans une mémoire non-volatiles. Pour chaque échantillon, vous avez 8 octets de données (la température et le temps).
2. Votre système embarqué affiche une page web avec plusieurs images et des informations qui varieront peu, mais qui seront lues fréquemment.
3. Vous échantillonnez du son à 20kHz – 2 octets par échantillons et vous voulez sauvegarder plusieurs bandes sonores de 2 minutes maximum pour les relire ensuite. Votre système a assez de mémoire RAM pour stocker 8Mo d'information.

Q4.7 Le MMU et le contrôleur de DMA sont étroitement reliés ensemble. Expliquez pourquoi.

Q4.8 Comment se fait le DMA dans le microcontrôleur des laboratoires? Quelles sont les contraintes d'accès à la mémoire de données par le cœur du microcontrôleur lors d'un transfert par DMA?

Q4.9 Quelle est la différence principale entre les mémoires DRAM et FRAM?

Q4.10 Il faut interfacer une mémoire de 1 méga-octet sur un bus d'adresses 32 bits (on assume le bus A31-A0 avec A31 MSB). La première adresse de la mémoire doit être 0x10E08000. Quelle est la fonction logique permettant d'activer la mémoire en mettant le bit 'ENABLE' de la mémoire à '0' à partir des lignes du bus d'adresse?

Q4.11 Combien de temps un bit de mémoire FLASH garde-t-il sa valeur lorsque la mémoire n'est pas alimentée?

Q4.12 Le contenu des mémoires non-volatiles est habituellement protégé par un CRC ou un checksum. Pourquoi? Quels événements peuvent corrompre les données ou instructions contenues dans ces mémoires?

Cours 5 : Périphériques, Général + GPIO

Q5.1 Lorsque deux composantes opèrent à des vitesses différentes, il faut un tampon de données et des drapeaux d'état pour gérer la communication entre les deux composantes. Expliquez à l'aide d'un exemple.

Q5.2 Quelles adresses faudra-t-il lire et/ou écrire afin de mettre un "1" sur la broche 29 d'un microcontrôleur STM32F40x, sachant que le package du microcontrôleur est LQFP100?

Q5.3 Dites pourquoi il y a habituellement un microcontrôleur juste pour contrôler un LCD dans un système microprocesseur et interfaces.

Q5.4 Expliquer comment fonctionne l'interface entre un contrôleur de LCD comme le HD44780 et un autre microprocesseur (le LCD du laboratoire 3 a un contrôleur HD44780).

*Q5.5 Combien de broches de microprocesseur sont nécessaires afin de contrôler un clavier ayant 72 touches? Pourrions-nous relier chaque broche à une touche? Si on assume que le clavier est une matrice 8*9, serait-il possible de lire le clavier avec 14 broches? Comment?*

Q5.6 Illustrez une sortie totem pole avec des transistors (BJT ou CMOS au choix) et illustrez une sortie open drain (ou open collector, au choix).

Q5.7 Dessinez le circuit pour qu'une LED nécessitant 10mA soit allumée avec un "0" lorsque la sortie du microprocesseur ne peut fournir que 4mA.

Q5.8 On veut lire un "1" lorsqu'un bouton est appuyé et un « 0 » sinon. Si le contact du bouton oscille énormément et si un debounce matériel est requis, dessinez le circuit permettant de lire ce bouton avec l'entrée digitale d'un microcontrôleur.

Q5.9 Quel problème peut survenir s'il n'y a pas de résistances de pull-up ou de pull-down sur les lignes de lecture d'un clavier?

Q5.10 Lors de l'interfaçage d'un LCD, dans quel cas utilise-t-on l'instruction de lecture du busy flag plutôt que d'attendre un délai défini?

Q5.11 Nommer les trois sortes de registres qu'on retrouve généralement permettant de contrôler les GPIO et à quoi sert chacun?

Q5.12 Pourquoi le registre de direction est toujours configuré comme des entrées à un reset?

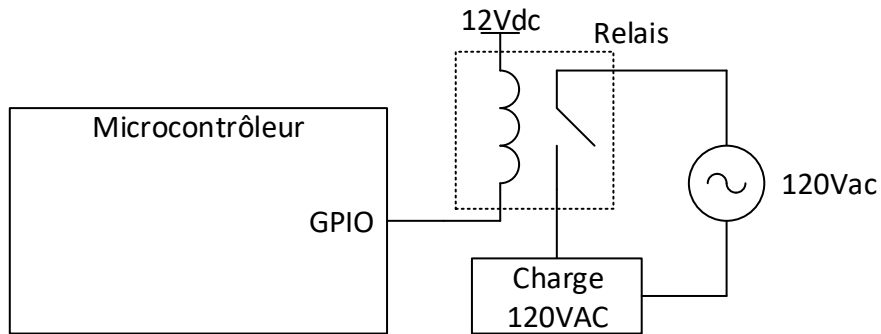
Q5.13 Pour quelle raison un condensateur peut être mis en parallèle avec une entrée (connecté entre l'entrée et la masse)?

Q5.14 Votre ami arrive avec un petit écran tactile et vous demande si son écran tactile est capacitif ou résistif. Comment ferez-vous pour lui donner une réponse juste?

Q5.15 Pourquoi utilise-t-on souvent deux circuits intégrés différents afin de contrôler les gros LCDs TFT (Thin-film-transistor liquid-crystal display)?

Q5.16 Quelles sont les différences entre une interface 8080 et 6800 pour le contrôle d'un LCD avec un bus parallèle?

Q5.17 Un étudiant veut contrôler un relais (Form A, Normally Open, Coil 12Vdc/150mA, Contact 120Vac/5A) avec son microcontrôleur semblable à celui des laboratoires. Il a donc branché son relais comme sur la figure ci-dessous. Ça ne marche pas. Quelles sont les erreurs de l'étudiant? Comment corriger le problème?



Cours 6: Périphériques, PWM, Timer, ADC et DAC

Q6.1 Le site web suivant détaille le fonctionnement des timers pour le 8051 d'Intel (le cœur de microprocesseur le plus vendu sur le marché mondial!) :

<http://www.mikroe.com/eng/chapters/view/65/chapter-2-8051-microcontroller-architecture/>.
Expliquez le fonctionnement du timer 0 du 8051.

Q6.2 Quels registres du STM32F407 doit-on écrire pour que le timer TIM2 déclenche une interruption à toutes les millisecondes? Quelles valeurs doit-on donner à ces registres si l'horloge du cœur fonctionne à 32MHz.

Q6.3 Énumérez quatre applications dans lesquelles on retrouve un PWM.

Q6.4 Peut-on convertir un signal de PWM en niveau DC dépendant des paramètres d'opérations du PWM? Si oui, comment? Si non, pourquoi? (Voir <http://www.esorol.com/assets/files/PDFs/Spectral%20analysis%20of%20a%20PWM%20signal.pdf> pour de l'aide!!!).

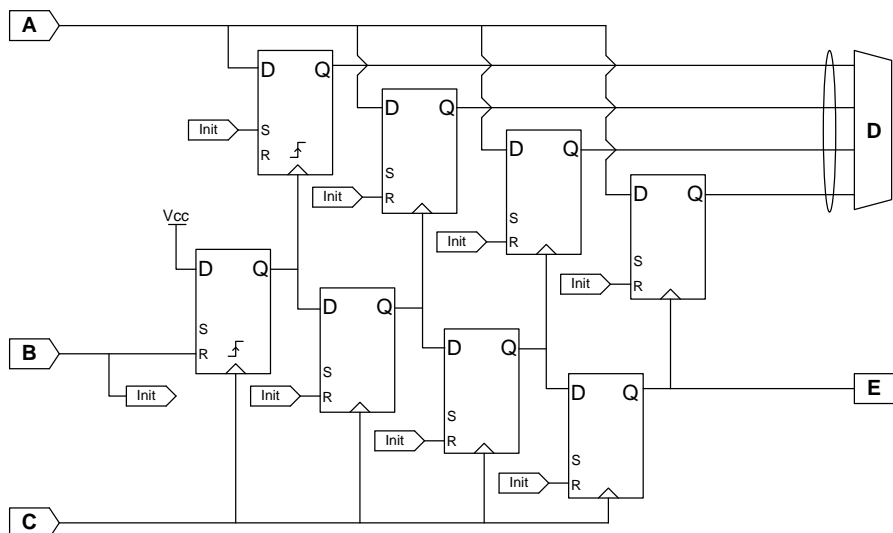
Q6.5 Peut-on produire un sinus avec un PWM? Comment?

Q6.6 Donnez un exemple d'application où la largeur des pulses d'un PWM est modulée.

Q6.7 Expliquez comment fonctionne un DAC R-2R.

Q6.8 Expliquez comment fonctionne un ADC par approximation successive?

Q6.9 La figure suivante illustre le « Successive Approximation Register (SAR) » d'un ADC 4-bits. Indiquez quels signaux internes de l'ADC sont branchés sur A, B, C, D et E?



Q6.10 Expliquez ce qui limite la vitesse d'échantillonnage et la résolution des DACs/ADCs. Utilisez le DAC R-2R et/ou l'ADC par approximations successives au besoin.

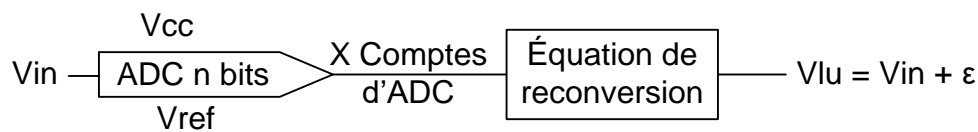
Q6.11 Pourquoi devrait-on toujours utiliser un filtre passe-bas entre le signal échantillonné et l'entrée d'un ADC lorsqu'on interface avec un ADC?

Q6.12 Décrivez les différentes manières d'utiliser un Timer?

Q6.13 Un timer 16 bits en mode auto-reload incrémente à tous les coups de l'horloge du timer. Sachant que l'horloge du système est 16MHz, sachant qu'il est possible de diviser l'horloge du système par 4, 8 et 16 afin de générer l'horloge du timer, quelle doit être la valeur de chargement automatique (auto-reload) du timer et quelle doit être l'horloge du timer pour générer une interruption à toutes les millisecondes. S'il existe plusieurs réponses possibles, donnez-les toutes!

Q6.14 Vous voulez utiliser un timer pour générer une interruption périodique. Malheureusement, ce timer n'a pas de mode auto-reload. Expliquez comment vous pourriez générer une interruption périodique en dépit de l'absence de l'auto-reload.

Q6.15 On retrouve, dans la littérature, plusieurs équations pour calculer la valeur mesurée par un ADC à partir de la lecture de l'ADC :



Parmi les équations de conversion suivantes, laquelle donnera, en moyenne, l'erreur de conversion la plus petite si l'entrée est totalement aléatoire?

- a) $V_{lu} = \frac{X \text{ counts}}{2^n \text{ counts}} * V_{ref}$
- b) $V_{lu} = \frac{X \text{ counts}}{(2^n - 1) \text{ counts}} * V_{ref}$
- c) $V_{lu} = \frac{(X + 0.5) \text{ counts}}{2^n \text{ counts}} * V_{ref}$
- d) Aucune de ces réponses

Q6.16 Lorsqu'un ADC 10 bits mesure 513 (valeur binaire retournée par l'ADC), quelle est la tension analogique à l'entrée de l'ADC, si on considère une référence de 2.5V et une alimentation de 3.3V?

Q6.17 Si on met 1.25V à l'entrée d'un ADC 12 bits ayant un voltage de référence à 3.3V : quelle valeur digitale donnera l'ADC (combien de comptes)? Lorsque le nombre de compte d'ADC sera reconvertit en valeur analogique pour calculer l'entrée, quelle sera l'erreur de mesure en fonction de la méthode de calcul (appliquer les trois équations de conversion énumérée précédemment)?

Q6.18 Un signal analogique haute-fréquence varie de -0.5V à 0.5V. Quel circuit électrique devrait être utilisé pour lire ce signal avec un ADC lisant de 0V à 5V (référence à 5V)?

Q6.19 Un DAC 8 bits a une tension de référence à 5V. Si on veut sortir 1.66V du DAC, comment de comptes doit-on mettre à l'entrée du DAC? Quelle est l'erreur par rapport à la sortie désirée?

Q6.20 La page 859 de http://www.atmel.com/images/atmel-11100-32-bit%20cortex-m4-microcontroller-sam4s_datasheet.pdf illustre le fonctionnement des timers su SAM4S, un microcontrôleur similaire à celui des laboratoires du cours. Détaillez le rôle de chaque section du contrôleur de timer qu'on retrouve sur la figure de cette page.

Cours 7: Périphériques, Interfaces Séries

Q7.1 *Quelle est la différence entre UART et RS232?*

Q7.2 *Quelle interface série est utilisée dans chaque application suivante :*

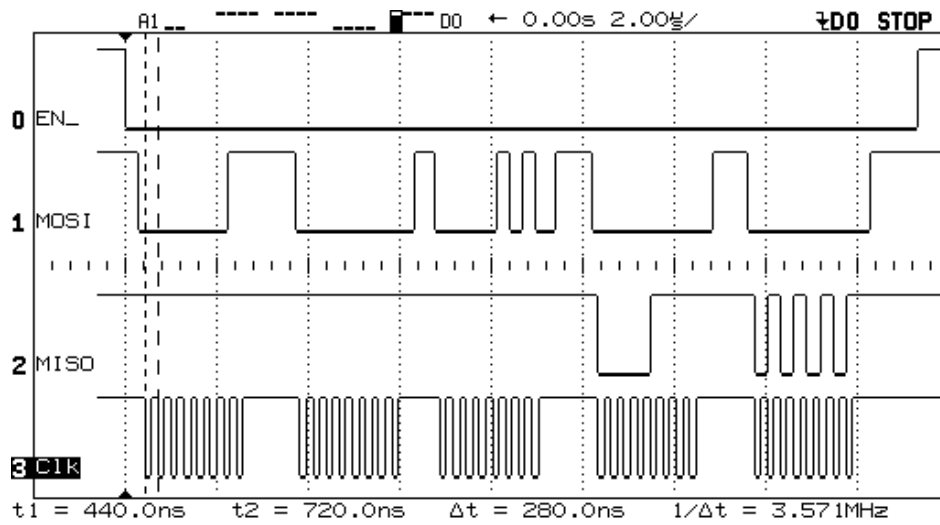
- a) *Cartes SD?*
- b) *Programmation JTAG?*
- c) *Circuit de contrôle des jets d'eau et alarmes d'incendies dans un immeuble à 30 étages?*
- d) *Mémoires séries?*
- e) *ADC/DAC externes?*
- f) *Système de communication dans un autobus.*
- g) *Système de surveillance dans une usine*

Q7.3 *Pour les quatre interfaces séries suivantes, donnez les caractéristiques générales :*

<i>Caractéristique</i>	<i>RS232</i>	<i>RS485</i>	<i>SPI</i>	<i>I2C</i>
<i>Architecture physique du réseau</i>				
<i>Architecture logique du réseau</i>				
<i>Sens de la communication</i>				
<i>Synchrone ou Asynchrone</i>				
<i>Vitesse max (bps)</i>				
<i>Distance max (m)</i>				
<i>Nombre de fils minimum</i>				
<i>Type de sortie requise pour écrire un bit</i>				
<i>Matériel de support requis</i>				
<i>Décennie de création</i>				

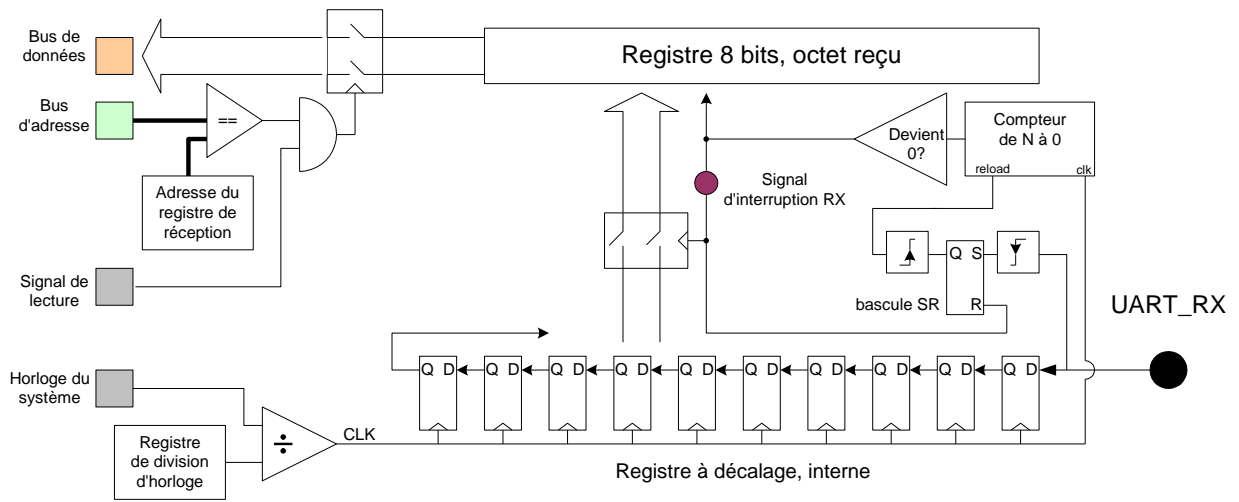
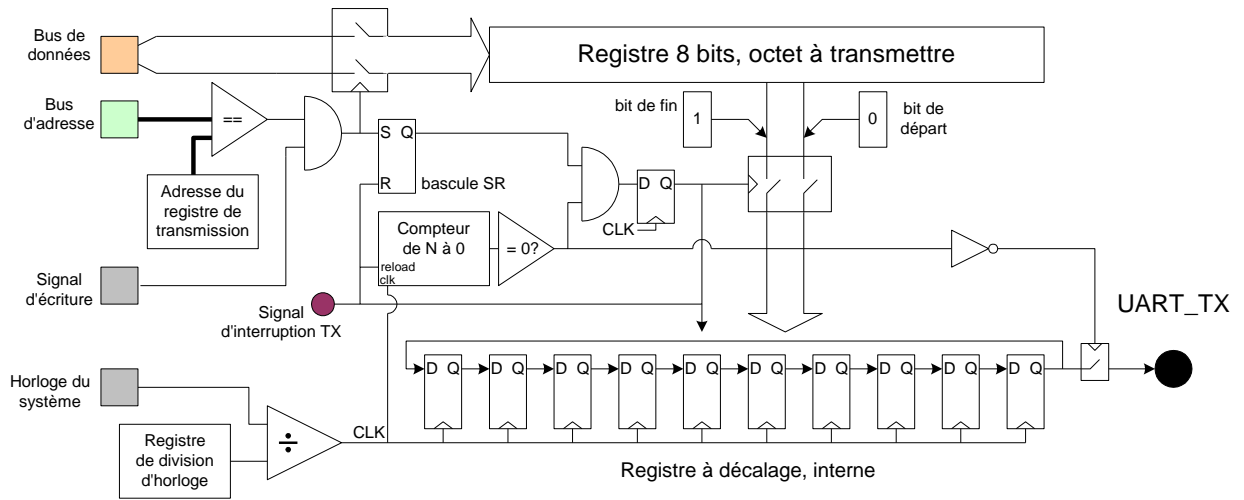
Q7.4 *Le RS485 utilise parfois 4 fils pour la communication, mais souvent 2 : il faut moins de fils! En contrepartie, la communication sur deux fils est HALF-DUPLEX, différentielle et bidirectionnelle. Expliquez pourquoi la plupart des systèmes de communication RS485 ont une architecture maître-esclaves en raison de ces caractéristiques.*

Q7.5 La figure suivante illustre une communication SPI entre un maître et son esclave.
Répondez aux questions reliées à la figure :



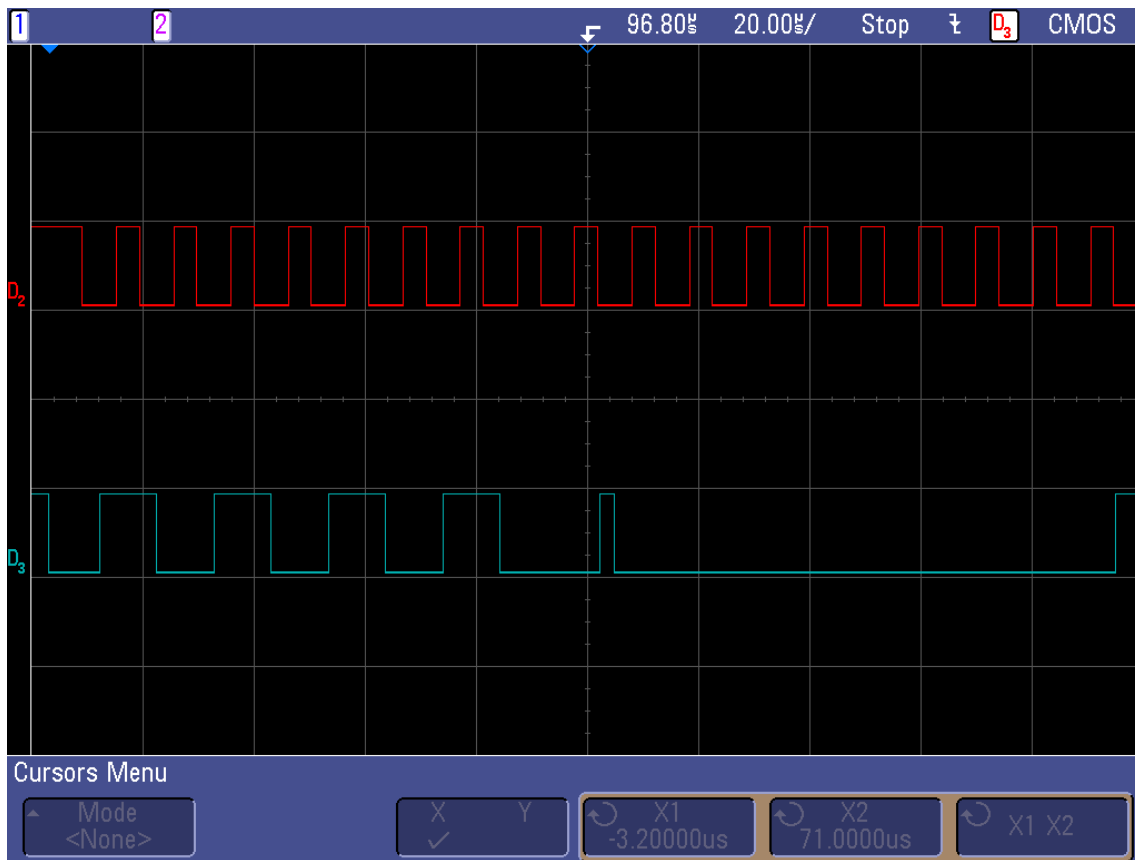
- Qui génère les différents signaux de communication? Le maître ou l'esclave?
- Combien d'octets sont transmis et reçus lors de la communication? Quelle est la valeur de ces octets?
- À quelle fréquence sont transmis les bits lors de cette communication SPI? Quelle est la vitesse de communication effective?
- Quelles sont la phase et la polarité du signal SPI?

Q7.6 La figure suivante illustre un module d'E/S pour un UART¹. À partir de cette illustration de principe, expliquez le fonctionnement général du périphérique :



¹ Le module d'UART présenté dans ces exercices est fictif. Malgré quelques erreurs possibles (par exemple, le nombre de bits du registre à décalage dans un UART est souvent plus petit que 10 bits!) et malgré le nombre très limité de caractéristiques qu'il illustre, ce dessin permettra au lecteur de comprendre le fonctionnement global d'un UART.

Q7.7 La capture d'oscilloscope suivante représente le début d'une communication I²C (le signal d'horloge commence au début du chronogramme). Le signal D2 est l'horloge (SCL) et le signal D3 le canal de données (SDA). Vu horizontalement, la première moitié du signal contient l'octet de contrôle, et la seconde moitié le début des données.



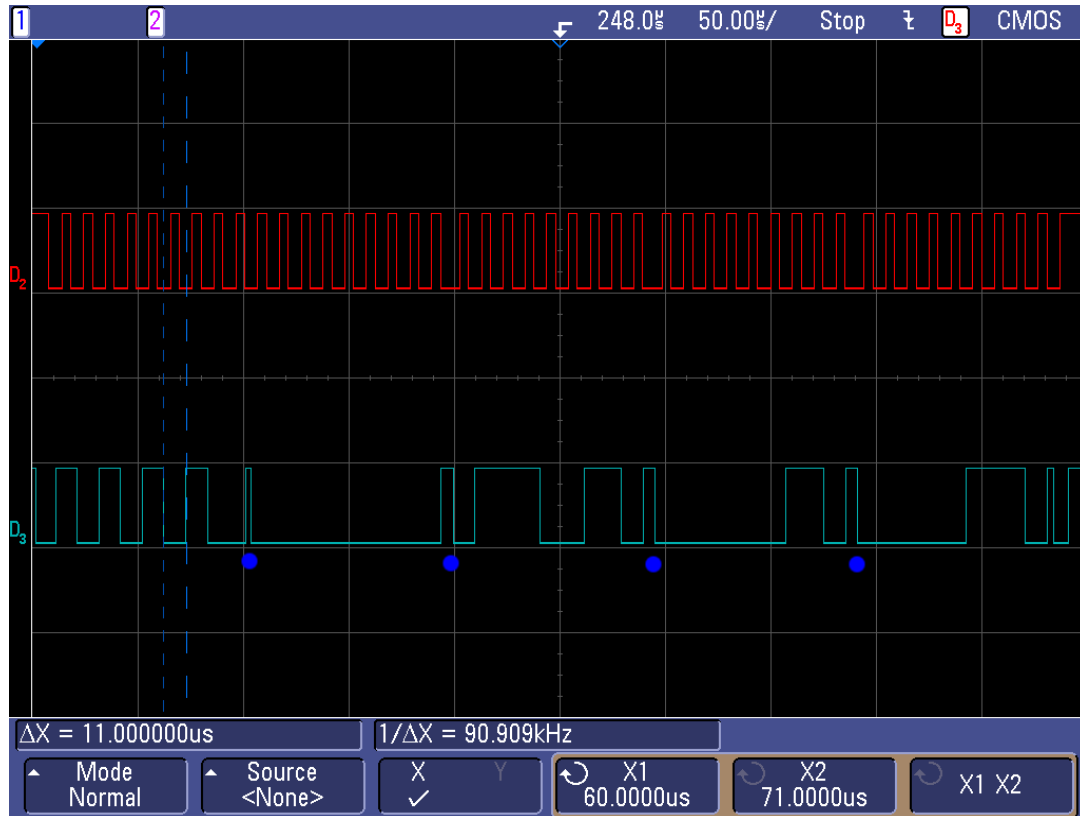
- Est-ce le maître ou l'esclave qui initie la communication? Comment?
- Quelle est l'adresse du périphérique visé par la requête?
- Considérant la taille de cette adresse, combien de périphériques peuvent-ils être adressés sur un bus I²C?
- Après l'envoi du premier octet de synchronisation, est-ce que c'est le maître ou l'esclave qui écrit sur le bus?

Q7.8 Pour lire un octet sur une mémoire EEPROM en I²C, le protocole est le suivant :

- Envoi d'un octet de contrôle pour sélectionner l'esclave
- Envoi de l'adresse (2 octets)
- Envoi d'un autre octet de contrôle sélectionnant le même esclave
- Lecture des données à partir de l'esclave

- Pourquoi l'octet de contrôle est-il envoyé deux fois alors que le canal de communication ne change pas (e.g. c'est le même esclave qui reste activé)?
- Pour chacun de ces octets envoyés, qui générera le bit de ACK? Qui générera l'horloge?

Q7.9 La capture d'écran suivante représente l'écriture de deux octets dans une mémoire de type flash (EEPROM). La mémoire est initialement vide (toutes les pages ont été effacées). Afin de faciliter la lecture du chronogramme, la largeur d'un bit est identifiée par les deux curseurs, et le début de chacun des octets de données est identifié par un point bleu. Le premier octet est l'octet de contrôle, les deux suivants représentent l'adresse (le premier octet contenant les MSB), et les deux derniers les données envoyées.



- Quelle est l'adresse à laquelle les données seront écrites?
- Si on fait par la suite une lecture à l'adresse 0x0074 (116 en décimal), quel sera le premier octet reçu? Le deuxième?
- D'après ce que vous savez du protocole I²C et de son implémentation matérielle, que représentent les petites transitions 0 → 1 → 0 sur SDA, que l'on peut par exemple voir au-dessus du premier point bleu.
- Tenant compte du temps de transmission d'un bit (identifié par les curseurs), et en négligeant le temps requis pour l'envoi de l'octet de contrôle, quelle est la vitesse de transmission effective (en octets/s) de ce lien I²C?

Cours 8, RTC, temps et interfaces parallèles

Q8.1 Qu'est-ce qu'un RTC? Pourquoi utiliser un RTC au lieu d'un compteur interne?

Q8.2 Combien d'octets au minimum sont nécessaires pour représenter une date de l'année actuelle à la milliseconde près?

Q8.3 Nommez deux avantages et deux inconvénients à utiliser une interface parallèle au lieu d'une interface série.

Q8.4 Décrivez le temps d'accès, le temps de cycle, la latence et la bande passante lorsque ces mots sont employés dans un contexte d'interface parallèle?

Q8.5 Pourquoi est-il important de donner la latence d'un périphérique?

Q8.6 À quoi servent habituellement les broches ALE, nCS, nOE et nWE d'une interface parallèle.

Q8.7 L'interface de communication avec une mémoire NAND-FLASH est souvent une interface utilisant essentiellement 8 lignes pour les adresses et les données. Comment peut-on adresser une mémoire NAND-FLASH de 1Mo?

Q8.8 Les interfaces parallèles servent souvent à communiquer avec des écrans ou des LCDs. Pourquoi?

Q8.9 Si vous utilisez une interface parallèle pour accéder à une mémoire SDRAM, quelle instruction exécutée par le microcontrôleur permettra de contrôler les broches nRAS et nCAS?

Cours 9: Programmation de périphériques

Q9.1 Pourquoi un registre de statut existe-t-il pour les interruptions d'un périphérique?

Q9.2 Nommez et décrivez les modes de transfert habituels des contrôleurs de DMA. Donnez ensuite un exemple de situation où chaque mode de transfert est utilisé.

Q9.3 L'utilisation du DMA et l'utilisation de caches dans le système de mémoire amène un nouveau problème dans les SMIs. Quel est ce problème?

Q9.4 On retrouve le mauvais exemple suivant dans les notes de cours :

```
__irq void UART_ISR(void)
{
    if( UnOctetEstRecu() )
    {
        Tableau[NombreOctetsRecusUART] = OctetRecu();
        NombreOctetsRecusUART++;
    }
    ...
}

void Main(void){
    init();
    while(1){
        ...
        if(NombreOctetsRecusUART > 0)
        {
            OctetAtraiter = Tableau[0];
            TraiteOctet(OctetAtraiter);
            DécaleOctetsDuTableauVersIndice0();
            NombreOctetsRecusUART--;
        }
    }
}
```

Il est dit un peu plus loin que pour corriger l'exemple, on pourrait appliquer 3 solutions différentes : désactiver les interruptions, utiliser un sémaphore ou éliminer la section critique en dupliquant des ressources. Changez le code pour du mauvais exemple pour illustrer ces trois solutions.

Q9.5 Un timer est programmé en mode Auto-Reload pour générer une interruption basse priorité à toutes les 10 millisecondes. Dans le même programme, une interruption du UART survient lorsqu'un octet est reçu. Cet octet est traité dans la routine de l'interruption du UART pendant 2 millisecondes. Sachant que l'interruption du UART a une priorité supérieure à celle du timer, répondez aux questions suivantes :

1. Si une interruption de UART se produit juste avant l'interruption de timer, qu'arrive-t-il avec l'interruption du timer?
2. Quand surviendront les prochaines interruptions de timer par rapport à celle avant l'interruption du UART?

Q9.6 Pour quelle raison le code suivant peut-il conduire à une attente, dans la fonction Wait, beaucoup plus longue que prévu? Comment corriger simplement la fonction?

```
void Main(void)
{
    ConfigureTimerToGenerateIntAtEveryMS();
    ConfigureNVICToGenerateTimerInt();
    while(1)
    {
        ...
        if(Event())
        {
            Wait(100);
        }
    }
}

void Wait(int TimeToWait)
{
    DecrementingCounter = TimeToWait;
    while(DecrementingCounter != 0)
    {};
}

int DecrementingCounter;

void Timer_ISR(void)
{
    DecrementingCounter --;
}
```

Q9.7 Pour la question précédente, proposez au moins une autre implémentation de la fonction Wait. Vous pouvez également changer le code dans Timer_ISR.

Q9.8 (pour experts!) Dans le code suivant, on veut entreprendre une action 5 millisecondes après la réception d'un octet sur le SPI : on considère que le message est fini et on le traite. Dans le système, une interruption de systick timer se déclenche périodiquement à toutes les millisecondes. Par ailleurs, une interruption de SPI se produit lorsqu'on reçoit un octet. Lors de la réception, on note aussi le temps. Toutes les interruptions ont la même priorité. Voici le code:

<i>Boucle principale</i>	<i>Interruption de SPI</i>	<i>Interruption de timer</i>
<pre>void main(void){ while(1){ if(MessageUnderRx) { if(MScount - LastRxTime > 5) { TraiteMessageSPI(); MessageUnderRx = 0; }}};</pre>	<pre>volatile unsigned int LastRxTime; void SPI_Rx_ISR(void) { MessageUnderRx = 1; LastRxTime = MScount; AjouteOctetRecuAuMessage(); }</pre>	<pre>volatile unsigned int MScount; void Systick_Ms_ISR(void) { MScount++; }</pre>

Malheureusement, le code n'attend pas toujours 5 millisecondes après la fin d'un message avant le traiter. À de rares occasions, il traite un message en plein milieu de sa réception. Bizarrement, le phénomène se produit lorsqu'on compile avec un compilateur, mais pas avec un autre compilateur. Sachant que les deux compilateurs compilent correctement (l'erreur est dans le code), pourquoi le problème survient-il?

Q9.9 Pour quelle raison l'accès au contrôleur SPI est parfois programmé directement, sans utiliser d'interruption ou de de DMA comme illustré ci-dessous?

```
void main(void)
{
    ...
    ConfigureSPIPourEtreMaitreEtParlerAvecSlaveSPI();
    while(1)
    {
        ...
        if(JeDoisEnvoyerUnOctetSPI())
        {
            EnvoieOctetSurSPI(OctetAEnvoyer);
        }
        ...
    }
}

void ConfigureSPIPourEtreMaitreEtParlerAvecSlaveSPI(void)
{
    REGISTRE_CONTROLE_SPI = VALEUR_POUR_ETRE_MAITRE;
    REGISTRE_CONTROLE_SPI += VALEUR_DE_POLARITE_ET_PHASE_DESIREE;
    REGISTRE_PRESCALE_SPI = VALEUR_POUR_FREQUENCE_SCLK_DESIREE;
    REGISTRE_CONTROLE_PERIPHERIQUE = VALEUR_POUR_ACTIVER_SPI;
    REGISTRE_ROLE_DES_BROCHES = VALEUR_POUR_UTILISER_BROCHES_POUR_SPI;
}

void EnvoieOctetSurSPI(unsigned char OctetTX)
{
    while(REGISTRE_STATUT_SPI != PRET_A_TRANSMETTRE)
    {};
    REGISTRE_TRANSMISSION_SPI = OctetTX;
}
```

Q9.10 En utilisant du pseudo code comme dans la question précédente, indiquez comment on pourrait refaire l'accès au SPI en utilisant des interruptions, c'est-à-dire sans attendre dans la fonction EnvoieOctetSurSPI.

Cours 10, Production et environnement

Q10.1 Expliquez pourquoi l'affirmation suivante est fausse : un ingénieur en électronique doit livrer ses appareils rapidement et, de ce fait, consacre la plus grande partie de son temps à implémenter des programmes embarqués ou implémenter des circuits électroniques.

Q10.2 Donnez au moins six (6) raisons qui peuvent pousser un ingénieur de production à changer une pièce recommandée par un ingénieur de recherche et développement.

Q10.3 Expliquez ce qu'est le JTAG.

Q10.4 Décrivez la norme RoHS et évaluez son impact en SMI.

Q10.5 Un manufacturier évalue la MTBF d'un modèle de disque dur à 90 000 heures. Un client se procure deux disques durs de ce modèle pour son serveur et les configure en redondance. Quelle est la probabilité que le montage fonctionne après 10 ans?

Q10.6 Les mémoires FLASH et EEPROM supportent un nombre maximum de cycles d'écriture. Nommez quelques stratégies que vous pouvez utiliser pour maximiser la vie utile de ces mémoires.

Q10.7 Nommez quatre stratégies pour limiter la consommation en puissance d'un système embarqué.

Q10.8 Quelle norme européenne établit les règles de mise au rebut des équipements électrique et électroniques?

Q10.9 Lorsque voulez tester la soudure des broches de deux circuits intégrés par JTAG, comment les broches TDO, TDI, TMS et TCK sont reliées à ces deux circuits?