Interruptions

GIF-1001 Ordinateurs: Structure et Applications, Hiver 2015

Jean-François Lalonde

Bref retour sur l'appel de fonctions

- 3 mécanismes importants:
  - détermination de l'adresse de retour
  - passage de paramètres & valeur de retour
  - préservation de l'environnement

Progression de PC

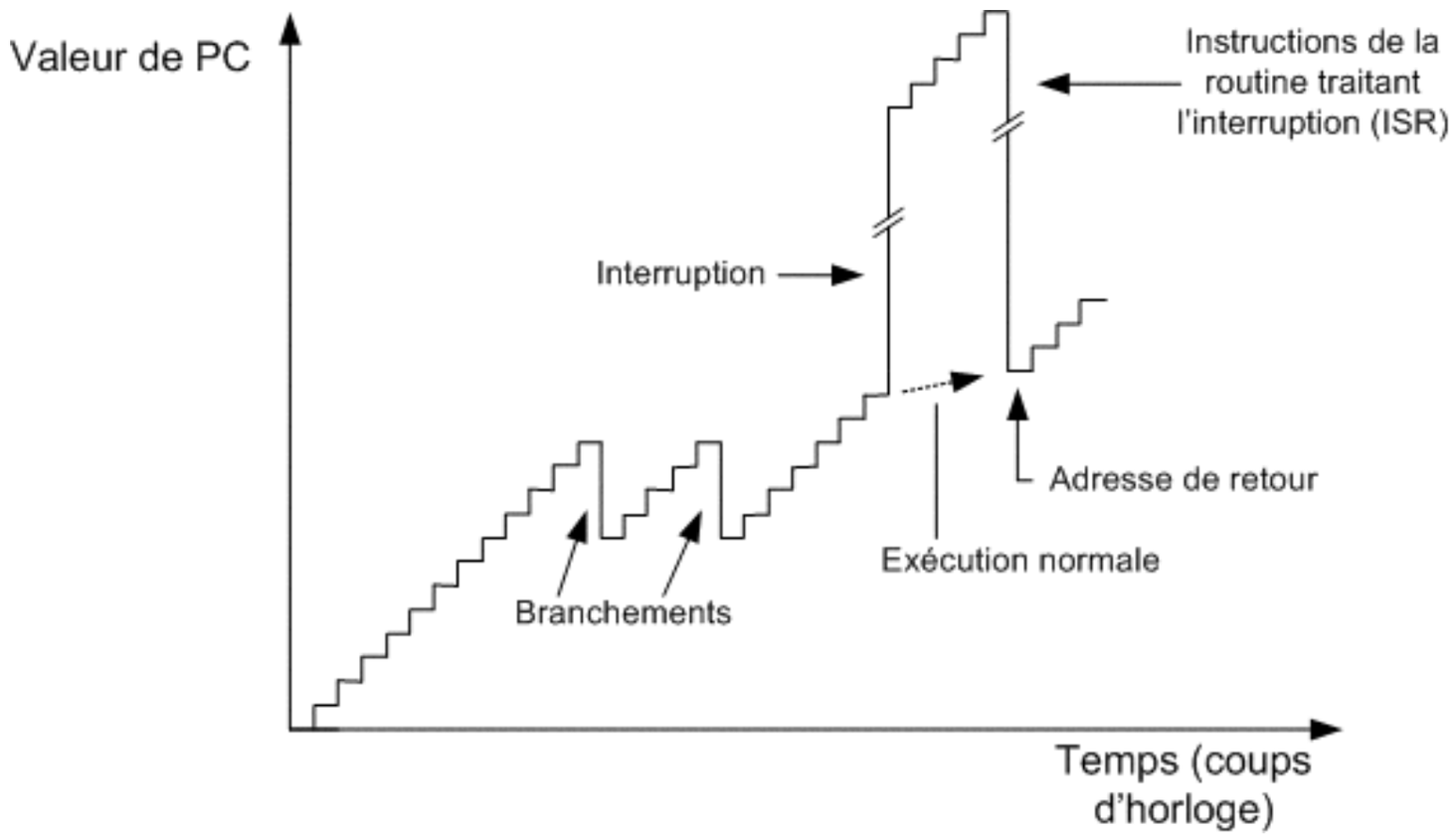
- Progression normale:
  - PC = PC + 4
- Branchements:
  - PC = adresse du branchement
- Appel de fonction:
  - PC = adresse de la fonction, LR = adresse de retour
  - PC = LR lorsque la fonction est terminée

Aujourd'hui: interruptions

- Pensons à un micro-processeur qui voudrait lire les valeurs écrites au clavier par un utilisateur.
- Comment faire?
  - Option #1: demander au clavier périodiquement s'il a reçu une nouvelle touche, sinon, attendre!
  - Option #2: c'est le clavier qui dit au micro-processeur qu'il a reçu une nouvelle touche!

Les interruptions

- Une interruption interrompt l'exécution des instructions par le microprocesseur.
- Lors d'une interruption:
  - l'exécution du programme principal est suspendue;
  - une sous-routine traitant l'interruption est exécutée;
  - puis le programme principal est continué.
- Différence entre interruptions et un branchement ou un appel de fonction?
  - les interruptions peuvent survenir n'importe quand pendant l'exécution.



Une interruption survient

- Terminer l'instruction en cours.
- Vérifier si l'interruption peut être traitée.
    - L'interruption peut être masquée (désactivée) ou ne pas être exécutée parce qu'une autre interruption de priorité supérieure ou égale est en cours. L'interruption est mise de côté pour être exécutée ultérieurement (lorsqu'elle sera réactivée ou lorsque l'interruption de priorité plus grande finira).
- Sauvegarder (sur la pile):
    - l'adresse de retour
    - les registres (en pratique, pas tous)
    - les drapeaux de l'ALU
- Déterminer l'adresse de la routine qui traitera l'interruption (Interrupt Service Routine, ou "ISR"). Cette routine est en mémoire code.
- Exécuter cette routine
    - PC = Adresse de l'ISR, donc l'exécution d'instructions continue dans l'ISR.

Routines de traitement d'interruptions

- Les routines de traitement d'interruptions ("Interrupt Service Routines — ISR) sont en mémoire
- Comment fait-on pour:
  - savoir quelle routine exécuter pour quelle interruption?
  - à quelle adresse est cette routine?
- Table des vecteurs d'interruption

Table des vecteurs d'interruption	
# d'INT	Contenu
0	Adresse de la routine à exécuter quand l'INT 0 survient
1	Adresse de la routine à exécuter quand l'INT 1 survient
2	Adresse de la routine à exécuter quand l'INT 2 survient
3	Adresse de la routine à exécuter quand l'INT 3 survient
…	…

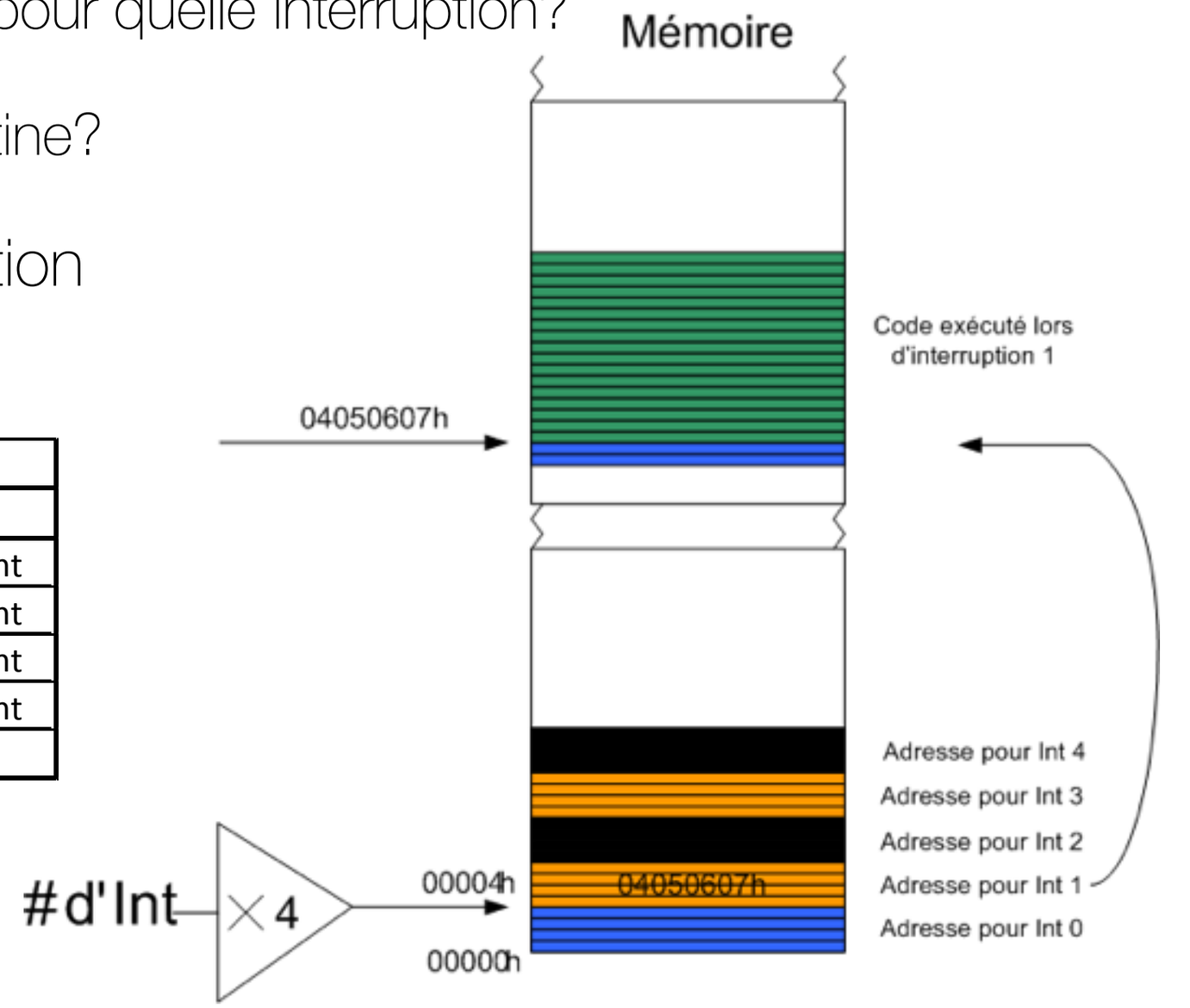


Table des vecteurs d'interruption

- Contient l'adresse de la sous-routine à exécuter lorsqu'une interruption survient.
    - Chaque entrée de la table est un "vecteur" qui mène aux instructions à exécuter pour traiter l'interruption.
- Commence habituellement à l'adresse 0h de la mémoire (ne pas mettre de code à cet endroit!).
- Habituellement en mémoire FLASH (contenu décidé par le compilateur/programmeur)
    - peut être déplacée en RAM
    - peut être modifiée par le système d'exploitation.

Interruptions imbriquées

- Qu'arrive-t-il si une interruption survient lorsqu'on traite une interruption?
- Cela dépend de la priorité
  - Si la priorité de la nouvelle interruption est plus élevée:
    - On interrompt l'exécution et on traite cette nouvelle interruption
  - Si la priorité de la nouvelle interruption est moins élevée:
    - On attend que le traitement de l'interruption à plus haute priorité soit terminé, et on traite cette nouvelle interruption par la suite
- Sauvegarder les adresses de retour et certains registres sur la pile permet d'imbriquer les interruptions comme on imbrique des fonctions

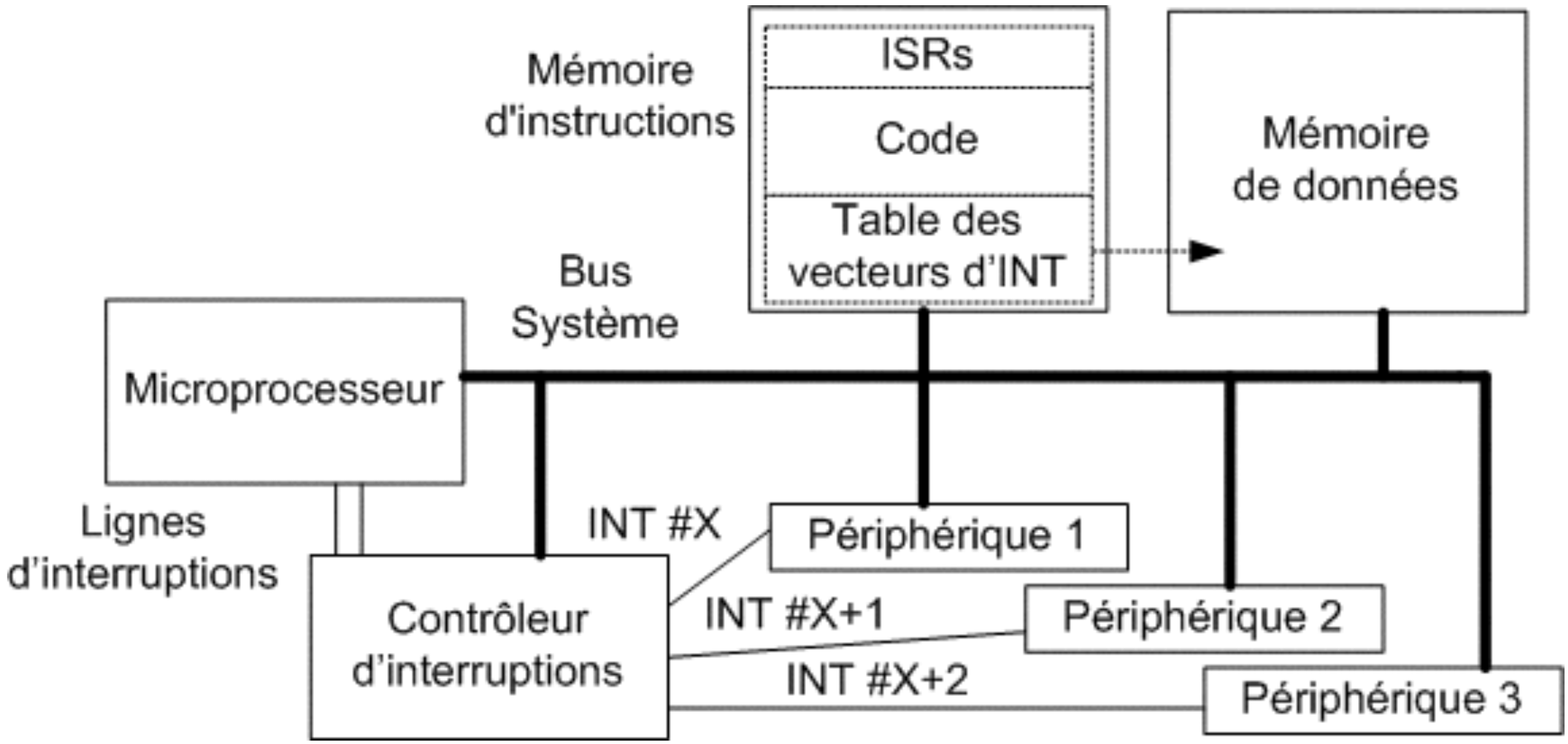
Une interruption se termine

- Une instruction est nécessaire pour indiquer la fin de l'interruption. Cette instruction, « retour d'interruption », est exécutée par le microprocesseur de la façon suivante:
  - Récupère les valeurs des registres sauvegardés sur la pile lors de l'entrée dans l'interruption
  - Récupère l'adresse de retour et les drapeaux sauvegardés sur la pile lors de l'entrée dans l'interruption
  - Met PC = adresse de retour de l'interruption

Contrôleur d'interruptions (NVIC)

- Les interruptions de l'ordinateur sont gérées par le contrôleur d'interruption.
- Le contrôleur d'interruptions:
  - reçoit les signaux d'interruptions
  - peut activer (masquer) ou désactiver certaines interruptions.
  - modifier la priorité des interruptions.
  - signale les interruptions au microprocesseur à l'aide de fils dédiés à cette fin.
  - peut être configuré via des instructions dans la mémoire
- Dans le cas du processeur ARM, le contrôleur d'interruptions est inclus dans le cœur.

Contrôleur d'interruptions



Types d'interruptions

- Système: reset, NMI (non-maskable Interrupt), faute matérielle générale, etc.

- Exception: le processeur peut générer des interruptions s'il n'est pas capable de lire ou d'exécuter une instruction (opcode invalide, division par 0, mémoire protégée, etc).

- Matérielles: générées par les périphériques

- Logicielles: il y a une instruction qui permet de générer une interruption dans tous les jeux d'instructions

- Il y a aussi des interruptions pour le mode "debug"...

Interruptions du système

- L'interruption reset est l'interruption système la plus commune. Cette interruption peut survenir pour plusieurs raisons:
    - mise sous tension, activation de la broche reset du microprocesseur, instruction reset, etc.
    - Lors d'un reset, toutes les autres interruptions sont ignorées
    - L'interruption NMI (Non Maskable Interrupt) est aussi fréquente dans les systèmes ordinateurs. NMI est une interruption que ne peut pas être désactivée par le logiciel et qui est souvent utilisée pour détecter une faute d'alimentation ou une mise hors tension.

Exceptions

- Les exceptions surviennent quand un évènement logiciel spécial arrive. Cet évènement logiciel empêche le microprocesseur d'exécuter l'instruction en cours pour diverses raisons:
    - instruction invalide, division par 0, référence à une adresse invalide, accès invalide à une adresse protégée, faute matérielle, etc.
- Les exceptions ont un très haut niveau de priorité parce le microprocesseur est dans une impasse: il ne peut exécuter l'instruction en cours en raison d'une erreur de programmation!

Interruptions matérielles

- Les interruptions matérielles sont générées par les périphériques
- La plupart des périphériques ont une ligne de contrôle reliée au contrôleur d'interruptions qui leur permet de signaler un événement.
- Lors d'une interruption de périphérique, le microprocesseur obtient automatiquement le # de l'interruption du NVIC et utilise ce numéro pour trouver l'ISR à exécuter à partir de la table des vecteurs d'interruptions.

Interruptions logicielles

- Les interruptions logicielles sont des interruptions « provoquées » par le programmeur. Le programmeur utilise une instruction qui déclenche une interruption.

- Les interruptions logicielles ont un effet similaire à un appel de fonction avec une différence fondamentale:
    - l'adresse de la fonction appelée est dans la table des vecteurs d'interruption plutôt qu'être une adresse relative au programme.

- Les interruptions logicielles servent souvent à appeler des fonctions du système d'exploitation dont l'adresse est inconnue du programmeur, mais gérée par le système d'exploitation (grâce à la table des vecteurs d'interruption).

Interruptions et système d'exploitation

- 2 utilités principales:
  - accès aux périphériques
  - exécution de plusieurs processus

Interruptions et système d'exploitation

- Accès aux périphériques
    - Chaque périphérique (ex: clavier, imprimante) peut se comporter légèrement différemment des autres
- C'est le système d'exploitation ("Operating System", ou OS) qui rend les programmes "indépendants" du matériel. Comment?
    - C'est l'OS qui modifie les ISR à exécuter en fonction du matériel branché dans l'ordinateur (via la table des vecteurs d'interruption)
- Les programmes peuvent donc utiliser la table des vecteurs d'interruption comme d'habitude

Accès aux périphériques

- Exemple: recevoir une valeur du clavier
  - interruption matérielle (c'est l'utilisateur qui tape au clavier)
- Exemple: envoyer des données à l'imprimante
  - interruption logicielle (c'est notre programme qui veut imprimer)
- Le système d'exploitation permet au même programme de "parler" à plusieurs modèles de claviers et d'imprimantes

Interruptions et système d'exploitation

- Les interruptions permettent l'exécution de plusieurs processus
- Comment?
  - Une horloge génère des interruptions périodiquement
  - À chaque interruption, on change le processus à exécuter

Priorités

- Que faire si une interruption survient lorsqu'une autre interruption est traitée?
- Les interruptions ont des priorités: une interruption haute-priorité peut interrompre une interruption ayant un niveau de priorité plus bas.
- Certaines interruptions peuvent survenir n'importe quand, même pendant une autre interruption.
- Certaines interruptions, comme reset, ont une priorité (maximale pour reset) qui ne peut pas être changée.
- Ordre de priorité (ARM)
  - Reset > système > exceptions > matérielles > logicielles

Interruptions ARM

- Entrée: Lors d'une interruption, le microprocesseur sauvegarde automatiquement les drapeaux, l'adresse de retour (PC), le registre de lien (R14), R12 et R3 à R0 sur la pile, dans cet ordre.

- La table des vecteurs d'interruption du processeur ARM est à l'adresse 0 de la mémoire. Il y a cependant possibilité de la relocaliser après le démarrage (en écrivant un registre spécial du cœur ou du NVIC). Le format et le contenu de la table des vecteurs d'interruptions dépend de la version de cœur ARM utilisée:

  - ARM7: La table des vecteurs d'interruption a 32 entrées de 4 bytes (128 octets au total). Chaque entrée de la table contient une instruction plutôt qu'une adresse ou vecteur. Cette instruction est habituellement une instruction de branchement… La dernière entrée de la table, réservée pour le FIQ (Fast Interrupt reQuest) ne contient pas de branchement, mais du code. Cela permet d'accélérer le traitement d'une interruption FIQ.

  - ARM Cortex M3: La table des vecteurs d'interruption contient des adresses, celles des routines de service des interruptions (ISR). La table a 64 entrées, toujours de 4 octets (256 octets au total). Il faut noter que la première entrée de la table (adresse 0) n'est pas un vecteur d'interruption, mais plutôt la valeur initiale de SP.

- Les interruptions ARM ont plusieurs niveaux de priorité et peuvent être masquées.

Interruptions ARM

- Sortie: Pour sortir d'une interruption, il faut mettre PC = 0xFFFFFFFX. Mettre PC à cette valeur, appelée EXCEPTION_RETURN cause un retour d'interruption. Le X, entre 0 et F, est un paramètre de retour de l'interruption.
    - Lors d'une sortie d'interruption, le microprocesseur dépile automatiquement R0 à R3, R12, LR, PC (adresse de retour) et les drapeaux.
    - Lors de l'entrée dans l'interruption, le microprocesseur met EXCEPTION_RETURN dans LR. La valeur d'EXCEPTION_RETURN est habituellement 0xFFFFFFFF. L'instruction permettant de retourner d'une interruption est généralement BX LR, comme lors d'un appel de sous-routine…
- Le cœur ARM a plusieurs mécanismes pour accélérer l'entrée et la sortie d'une interruption.
    - Les instructions sont courtes. Les instructions longues (store/load multiples) peuvent être interrompues. Finir l'instruction en cours est court.
    - Plusieurs opérations se font en parallèle lors de l'entrée dans l'interruption.
    - Si des interruptions se chevauchent, le processeur évite des empilements et dépilement inutiles.

Interruptions ARM et OS

- Pour les concepteurs de cœur ARM, toutes les interruptions sont gérées par le système d'exploitation. Le cœur a deux modes d'opération: normal (thread) et privilégié (handler). Le mode privilégié (handler) est entré automatiquement lors d'exécution d'interruptions. Certaines opérations du processeurs ne sont permises qu'en mode privilégié.

- Le cœur ARM implémente une minuterie (un timer) qui permet de générer des interruptions périodiquement. À toutes les X millisecondes, l'interruption du Systick timer peut survenir si le cœur est programmé à cet effet.

- Le microprocesseur ARM supporte les interruptions logicielles. L'instruction SVC #X (SuperVisor Call) permet d'appeler une interruption avec son numéro.