Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Systèmes Embarqués 1 & 2

p.02 - Interruptions (2e partie)

Classes T-2/I-2 // 2017-2018

Daniel Gachet | HEIA-FR/TIC

p.02 | 08.03.2018

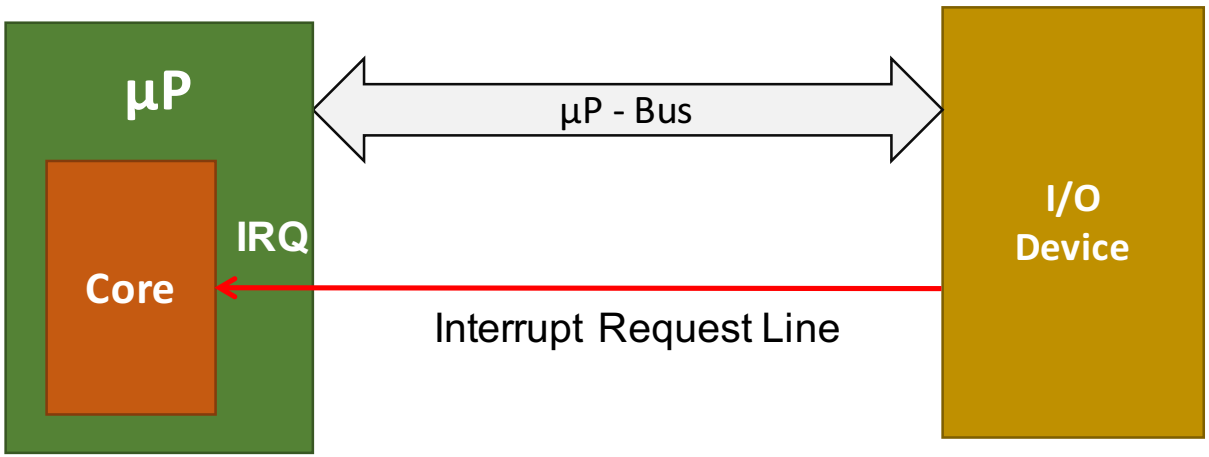
Hes·so
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Contenu

- Principe des interruptions matérielles
- Interruptions multiples
- Système d'interruptions des µP ARM Cortex-A8
- Système d'interruptions du µP TI AM335x

Principe

- Le périphérique signale sa requête d'interruption au µP en activant la ligne de requêtes d'interruptions *(Interrupt Request Line)*



- Lorsque le µP reçoit la requête
  - il termine l'exécution de l'instruction en cours
  - il sauve l'état courant (état de ses registres, cpsr et pc)
  - il exécute une routine de traitement d'interruption *(Interrupt Handler)* pour servir le périphérique
  - il retourne au programme interrompu après avoir servi le périphérique

Besoins

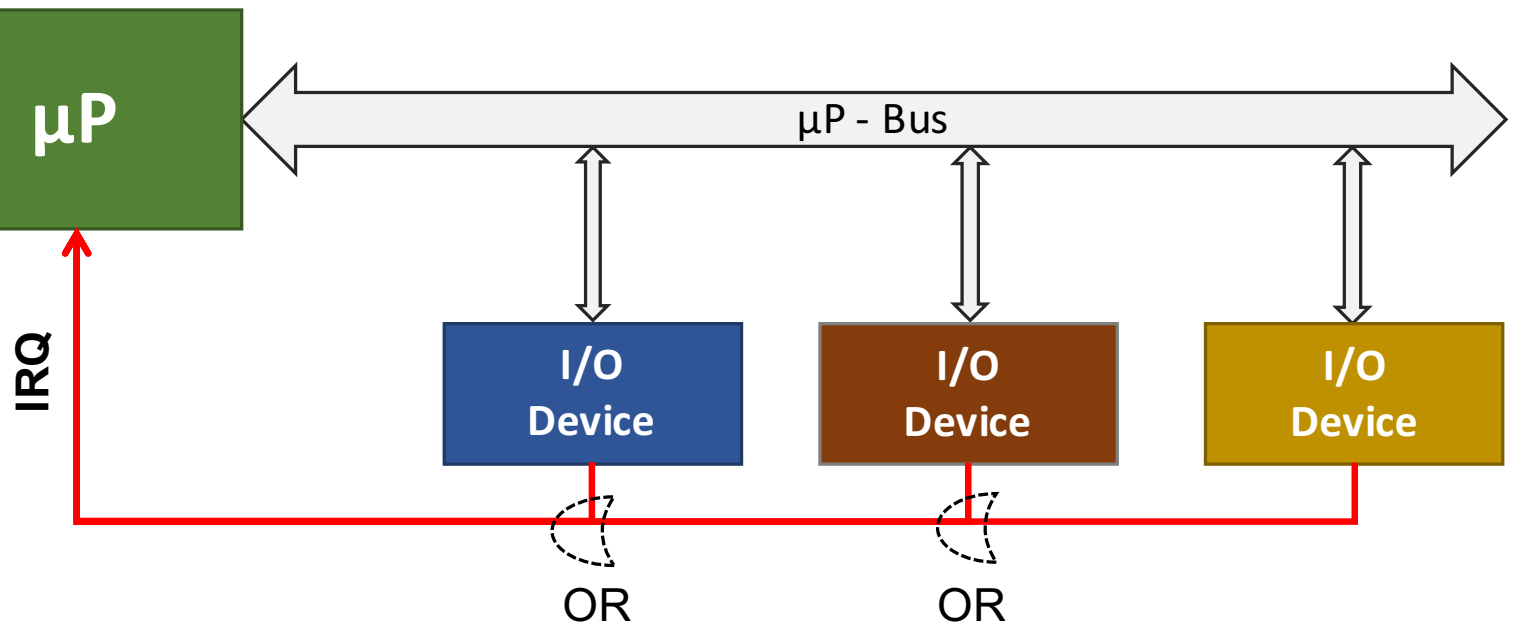
- Identifier le périphérique
  - Lorsque plusieurs périphériques sont connectés sur le même µP, le traitement des interruptions suppose que le µP puisse identifier le périphérique ayant levé l'interruption
- Arbitrer les requêtes multiples
  - Si plusieurs périphériques peuvent lancer simultanément une requête d'interruption, le µP doit être capable d'arbitrer les conflits d'accès en servant l'un après l'autre les périphériques demandeurs, ceci en fonction du degré d'urgence des besoins à satisfaire

Techniques

- Différentes techniques permettent au µP de déterminer la source de l'interruption
  - Scrutation logicielle
  - Priorité d'interruption
  - Interruption vectorisée

Scrutation logicielle

- Une seule ligne de requêtes d'interruptions pour tous les périphériques



Scrutation logicielle (II)

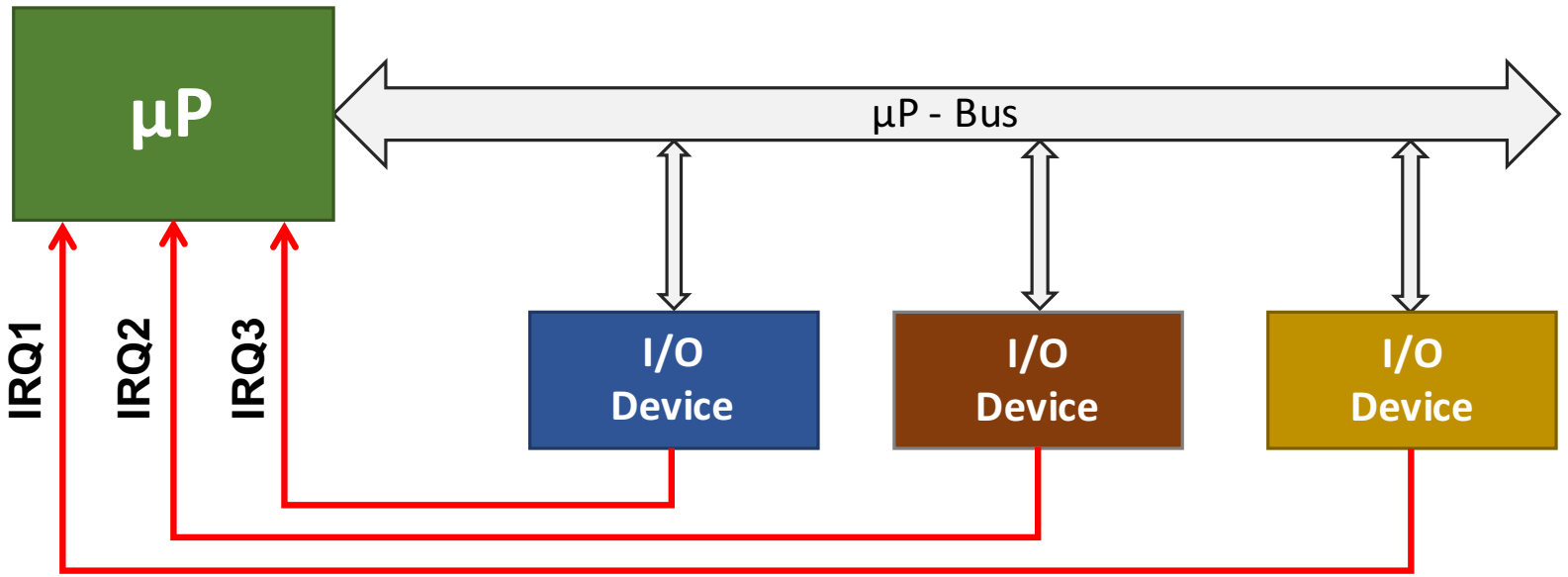
- Pour la scrutation logicielle if suffit d'une ligne unique de requêtes d'interruptions sur laquelle tous les périphériques sont branchés en parallèle pour former un OU câblé
- Les périphériques sont équipés de registres de contrôle et d'état
  - Ces registres permettent de gérer le périphérique ansi que de bloquer ou d'autoriser la levée d'interruptions par le périphérique
  - Ils permettent également d'identifier le périphérique ayant levé l'interruption
- Si une interruption apparaît, le µP suspend le programme en cours et exécute la routine de traitement d'interruptions
- Cette routine interroge successivement tous les périphériques pour déterminer celui qui a activé le signal d'interruptions

Scrutation logicielle (III)

- Deux méthodes de scrutation logicielle *(interrupt polling)*
  - Priorité fixe
    les périphériques sont scrutés dans un ordre précis et déterminé à l'avance *(fixed priority)*. Le 1er périphérique rencontré ayant levé une interruption est servi, puis le 2e et ainsi de suite...
  - Tourniquet
    les périphériqes sont ordonnés comme dans la première méthode, mais la recherche débute depuis le dernier périphérique servi *(round-robin)*. La recherche recommence au début, lorsque la fin de la liste est atteinte.
- La scrutation logicielle est simple à mettre en oeuvre, mais peut prendre énormément de temps selon le nombre de périphériques connectés au µP

Priorité d'interruption

- Une ligne de requêtes d'interruptions dédiée à chaque périphérique

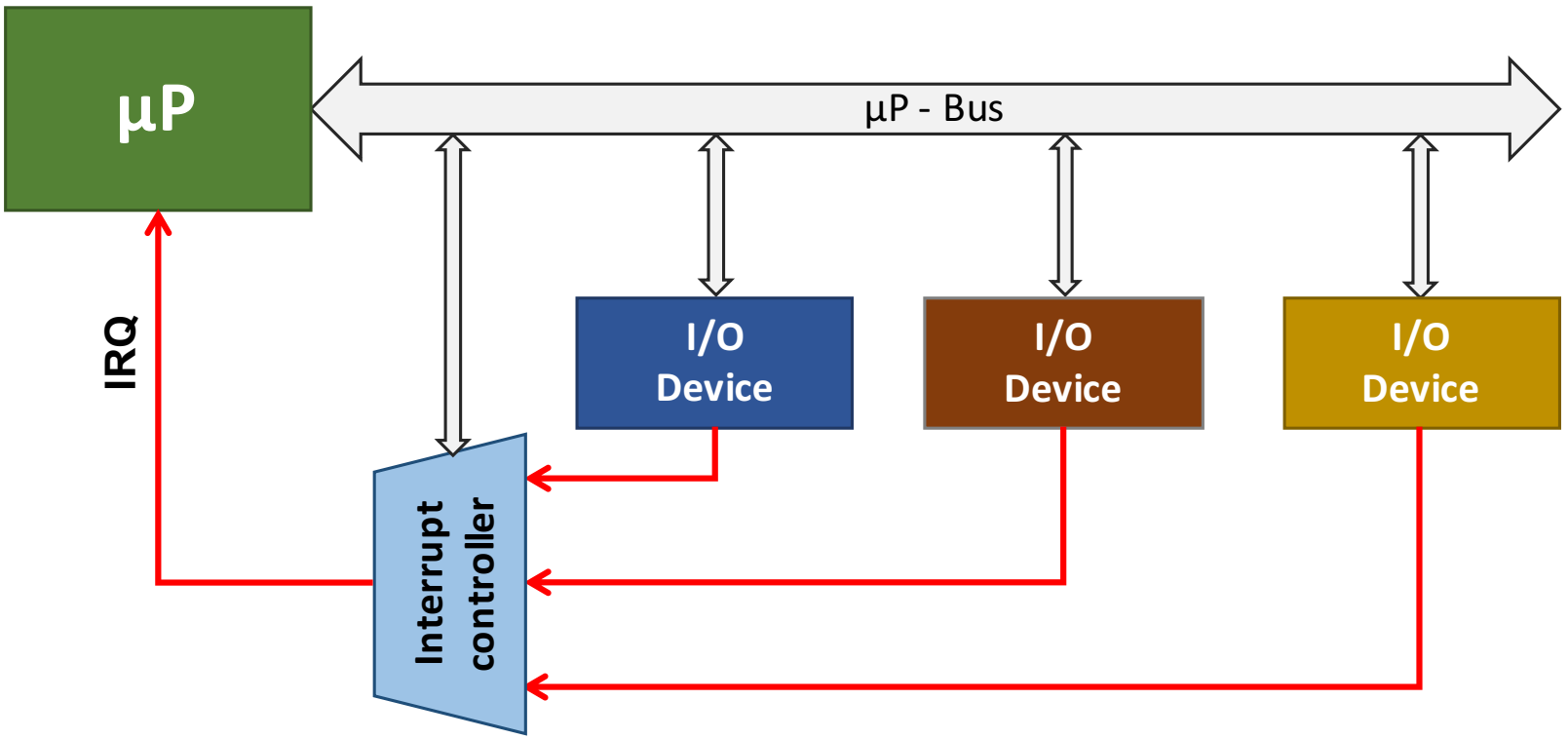


Priorité d'interruption (II)

- Afin de prioriser les interruptions, les µP mettent à disposition plusieurs lignes de requêtes d'interruptions
- Le µP affecte à chacune des lignes de requêtes d'interruptions une priorité matérielle fixe et unique, par exemple aux lignes IRQ1, IRQ2, IRQ3, le µP donnera les priorités 1, 2 et 3
- Chaque périphérique dispose de sa propre ligne de requêtes d'interruptions
- Lorsqu'une ou plusieurs requêtes d'interruptions apparaissent, le µP examine les lignes de requêtes d'interruptions pour déterminer la ligne active la plus prioritaire afin de servir le périphérique correspondant

Interruption vectorisée

- Un contrôleur capable de connecter, de prioriser et d'identifier plusieurs périphériques

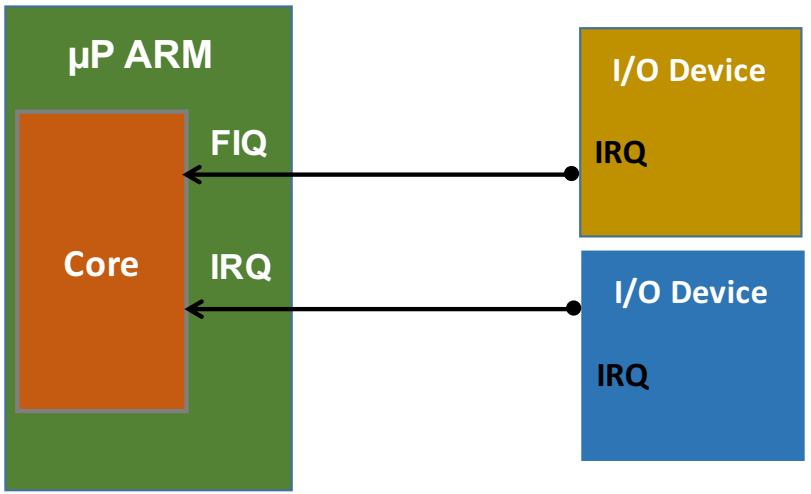


Interruption vectorisée (II)

- Une interruption vectorisée est définie par la capacité d'un contrôleur d'interruptions de connecter, d'identifier et prioriser une grande quantité de périphériques
- Chaque périphérique est connecté directement au contrôleur par une ligne de requêtes d'interruptions unique
- Le contrôleur dispose d'une logique interne pour prioriser et identifier les différentes sources d'interruptions
- Cette logique lui permet d'activer la ligne de requêtes d'interruptions du µP si un ou plusieurs périphériques ont signalé une interruption
- Lorsqu'une interruption est levée, la routine de traitement appelée par le µP obtient le vecteur d'interruption (l'identité du périphérique) en lisant un registre du contrôleur d'interruptions

ARM – Système d'interruption

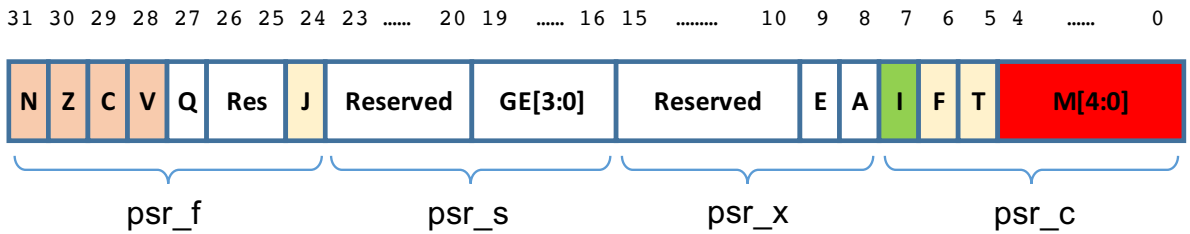
- Les µP ARM ne disposent que de deux lignes de requêtes d'interruptions, la Interrupt Request (IRQ) et la Fast Interrupt Request (FIQ)



- Ces deux lignes permettent de connecter très simplement des périphériques au µP ARM
- Chacune de ces lignes correspond à un niveau de priorité fixe et prédéfini

ARM – Acceptation des interruptions

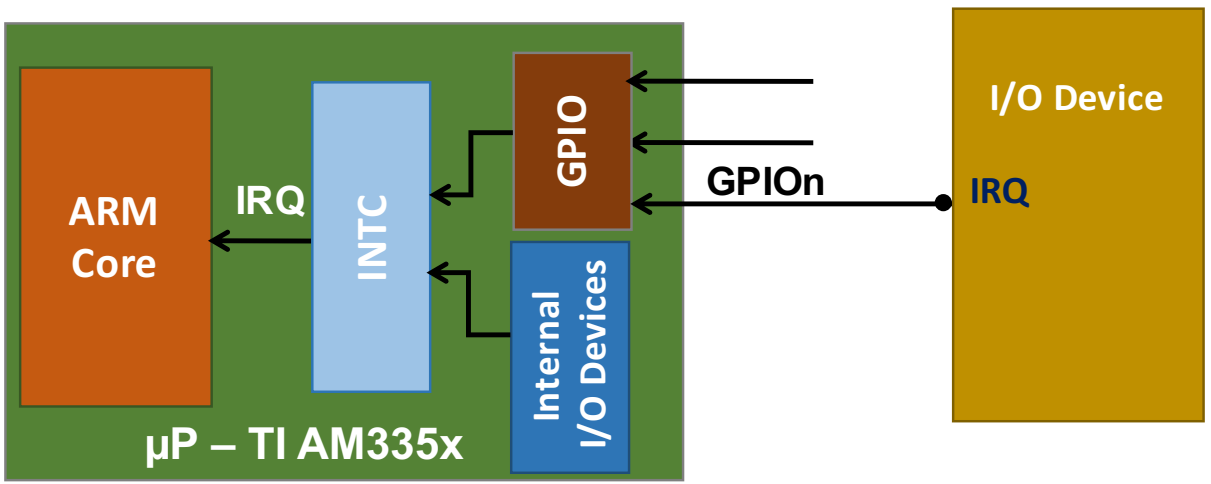
- Les bits I et F du registre CPSR offrent au µP la possibilité d'autoriser ou de bloquer le traitement des interruptions matérielles



- Le bit I permet d'autoriser les interruptions provenant de la ligne d'interruptions IRQ (I==0) et le bit F de la ligne FIQ (F == 0)
- Lorsqu'une interruption est levée, le µP change de mode et sauve son état dans les registres LR et SPSR, puis il empêche que des interruptions futures arrivent en mettant
  - I à 1 si une IRQ ou une FIQ a été levée
  - F à 1 si une FIQ a été levée
- Finalement, le µP poursuit le traitement de l'interruption en appelant la routine de traitement placée dans sa table de vecteurs

TI AM335x - Système d'interruptions

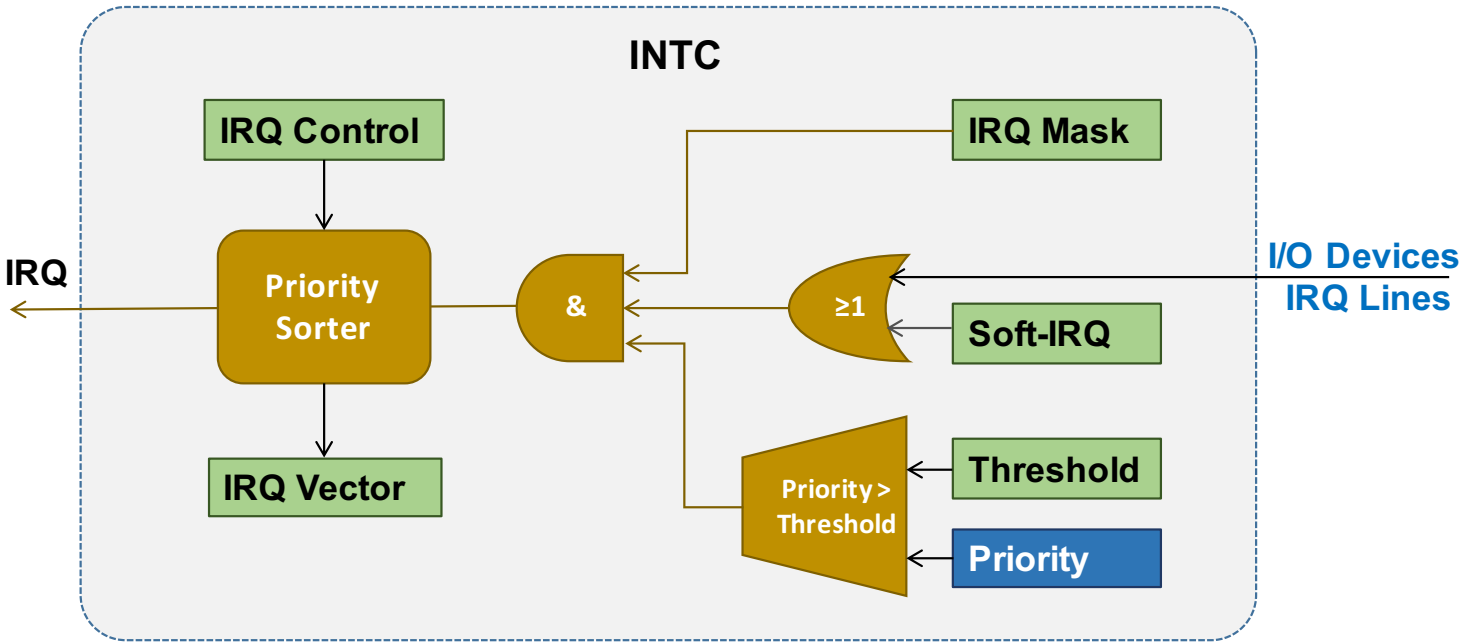
- Pour permettre d'interfacer avec un plus grand nombre de périphériques, le µP ARM TI AM335x dispose d'un contrôleur d'interruptions (INTC)



- Ce contrôleur est connecté au µP que par la ligne IRQ (pas de FIQ)
- L'INTC permet de collecter les requêtes d'interruptions de 128 sources différentes (périphériques internes au µP)
- Il est capable de prioriser ces sources et de générer un vecteur d'interruption pour la source la plus prioritaire

TI AM335x - INTC - Block Diagram

- Le contrôleur INTC permet de prioriser et de bloquer chaque ligne de requêtes d'interruptions, ainsi que de simuler par logiciel la levée d'une interruption



- Lorsque des interruptions sont levées, le contrôleur sélectionne, grâce à au niveau de priorité attribué aux sources d'interruptions, la source la plus prioritaire et la sert en premier

TI AM335x - INTC - Niveaux de priorité

- Le contrôleur INTC implémente un décodeur de priorités ne prenant en compte que les sources ayant un niveau suffisant pour être traitées
- Si le niveau de priorité de la source d'interruption est inférieur ou égale à une valeur prédéfinie, le threshold, la requête sera bloquée jusqu'à ce que le threshold soit ajusté

Threshold	Source Priority Level						
	0	1	2	3	...	126	127
0xff	√	√	√	√	√	√	√
0	x	√	√	√	√	√	√
1	x	x	√	√	√	√	√
2	x	x	x	√	√	√	√
3	x	x	x	x	√	√	√
...	x	x	x	x	x	√	√
126	x	x	x	x	x	x	√
127	x	x	x	x	x	x	x

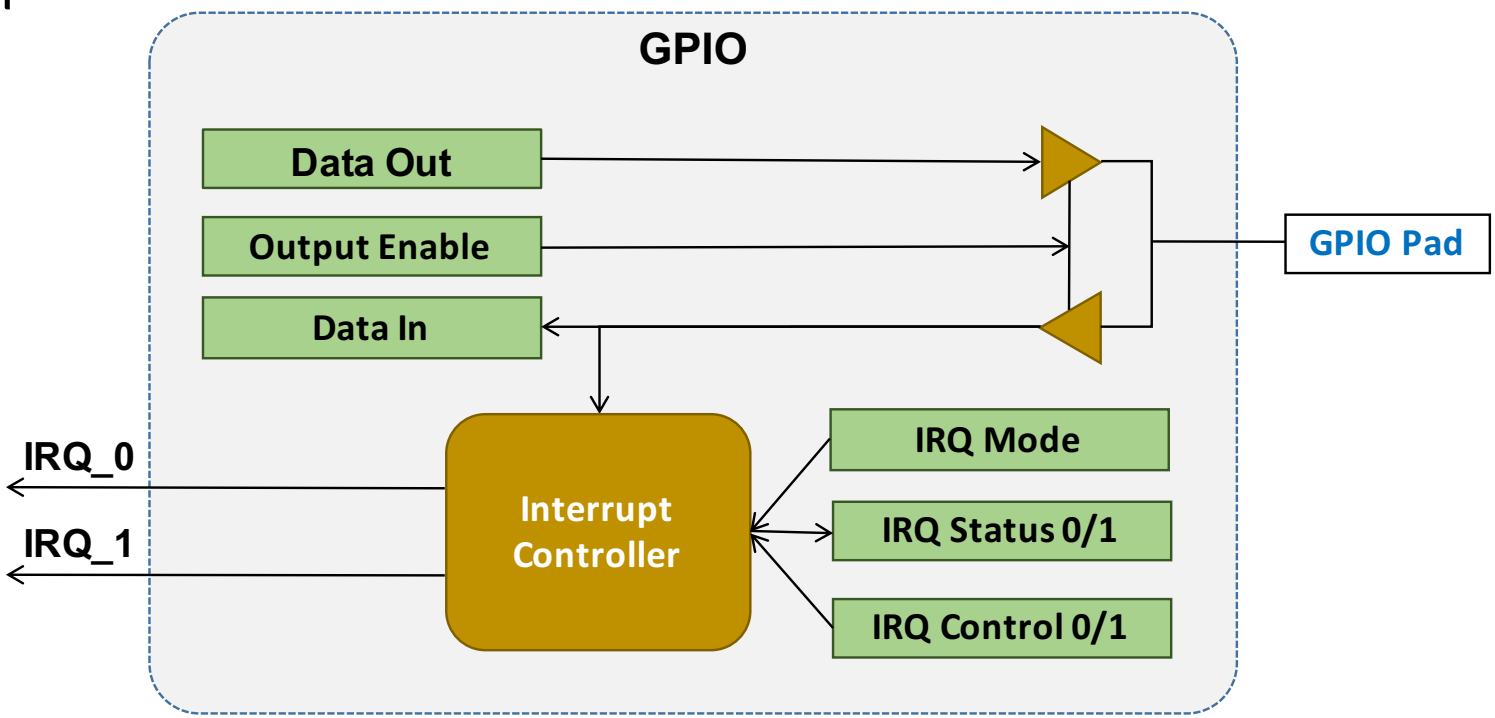
Current Threshold →

Unaccepted interrupts

Accepted interrupts

TI AM335x - GPIO - Block Diagram

- Le µP TI AM335x compte 4 ports GPIO (port 0 à port 3) avec chacun 32 broches (gpio-pads)
- Chaque broche peut être configurée en entrée ou en sortie
- Si la broche est configurée en entrée, elle peut également servir de ligne de requêtes d'interruptions pour des périphériques externes au µP



TI AM335x - GPIO - Configuration des interruptions

- Le module GPIO peut être configuré pour générer une interruption s'il détecte
  - un niveau haut du signal (high level)
  - un niveau bas du signal (low level)
  - un flanc descendant (falling edge)
  - un flanc montant (rising edge)

- Le module GPIO permet également de choisir pour chaque port si l'interruption à générer doit être sur la ligne 0 ou 1 (IRQ0 ou IRQ1)

ARM - Priorité des interruptions/exceptions

- La notion de priorité des interruptions/exceptions ne prend tout son sens que lorsque celles-ci surviennent simultanément
- Les µP ARM classe les exceptions sur 7 niveaux, le niveau 1 le plus prioritaire et le niveau 7 le moins

Priority		Exception
Highest	1	Reset
	2	Data Abort
	3	FIQ
	4	IRQ
	5	Imprecise Abort
	6	Prefetch Abort
Lowest	7	Undefined instruction / SVC / BKPT

- Le µP commence le traitement des interruptions par la plus prioritaire