

Exercice sur la pagination ARM – Corrigé

(v2017.2.1)

Soit une plate-forme équipée d'un processeur ARM 32-bit avec une RAM et un certain contenu, comme ci-dessous:

0xa4000000	
0xa0105ffc	0xa01decc2
0xa00100d4	0xa010881a
0xa0004e08	0xa0105093
0xa0003c00	0xa0010133
0xa0001100	0xa100007e
0xa0000000	

a) Quelle est la taille de la RAM ?

- $0xa4000000 - 0xa0000000 = 0x4000000 \Rightarrow 4 * 0x10'00000 \Rightarrow 4 * 16 \text{ Mo} = \underline{\underline{64 \text{ Mo}}}$

b) En supposant que la MMU (registre cp15/c2/c0) soit configurée pour pointer vers la table de page à l'adresse **0xa0001000**, que pouvez-vous déduire sur le mappage en cours (plages d'adresses virtuelles-physiques) en vous basant sur le contenu?

- On sait que la table de page de premier niveau se trouve à l'adresse **0xa0001000** et qu'elle contient obligatoirement 4'096 entrées. Chaque entrée faisant 4 octets (32 bits), la taille de la table fera ainsi 16 Ko ($0x4000$) $\Rightarrow$ La table occupe donc les adresses de **0xa0001000 – 0xa000'4fff**

- A l'adresse **0xa0001100** se trouve la valeur **0xa100007e**. Il s'agit clairement d'une entrée de la table de 1^{er} niveau. Les 2 derniers bits indiquent qu'il s'agit d'un mappage de type section. L'offset étant de 0x100, et chaque entrée occupant 4 octets, **l'index de niveau 1** vaut
$$\frac{0xa001100 - 0xa001000}{4} = \mathbf{0x40}.$$

⇒ Les adresses mappées sont les suivantes:

- Adresses virtuelles mappées: **0x04000000 – 0x040fffff**
- Adresses physiques mappées: **0xa1000000 – 0xa10fffff**
- Le contenu à l'adresse **0xa0003c00** correspond toujours à une entrée de la table de page de 1^{er} niveau. Elle se trouve à l'offset **0x2c00**, ce qui correspond à un index de niveau 1 égal à **0xb00**.
 - Les 2 derniers bits de l'entrée (**0xa0010133**) nous indiquent qu'il s'agit d'une table de type *Fine Page*. Les bits de poids forts de l'adresse physique seront égaux à la valeur **0xa0010**.
 - On sait que la table de second niveau se trouvera à cette adresse (début de la table). On trouvera donc à cet endroit une table devant occuper 1024 entrées au maximum (x 4 octets = 4'096 = 4 Ko = 0x1000 octets).
 - Une entrée possible de cette table est l'entrée à l'adresse **0xa00100d4**, où l'on trouve la valeur **0xa010881a**. Les 2 derniers bits de cette valeur indiquent un mappage vers une page de 4 Ko. Cette dernière se trouvera à l'adresse physique **0xa0108xxx**.
 - L'index de niveau 2 sera codé sur 10 bits (*fine page table*), et l'offset de la page est sur 12 bits. Il y a donc un recouvrement de 2 bits entre l'index et l'offset (les 2 derniers bits de l'index, les 2 premiers bits de l'offset). L'offset de l'entrée de niveau 2 vaudra
$$\frac{0xa00100d4 - 0xa0010000}{4} = \mathbf{0x35}.$$
 Les adresses mappées seront les suivantes:
 - Adresses virtuelles mappées: **0xb000d400 – 0xb000d7ff**
 - Adresses physiques mappées: **0xa0108400 – 0xa01087ff**
 - L'entrée à l'adresse **0xa0105ffc** ne pouvant appartenir à la même table de niveau 2 (offset > 0x1000), cette entrée appartient visiblement à une autre table de page.
- Le contenu à l'adresse **0xa004e08** correspond aussi à une entrée de type *Fine Page*. Le raisonnement est le même que le précédent.
 - Adresses virtuelles mappées: **0xf82ffc00 – 0xf82fffff**
 - Adresses physiques mappées: **0xa01dec00 – 0xa01defff**