QXD0133 - Arquitetura e Organização de Computadores II

Universidade Federal do Ceará - Campus Quixadá

Thiago Werlley
thiagowerlley@ufc.br

18 de outubro de 2025

Capítulo 13

Capítulo 13

Memory Protection Units - ARM System Developer's Guide

CP15 e Cache

Coprocessor 15 registers that configure and control cache operation.

Function	Primary register	Secondary registers	Opcode 2
Clean and flush cache	*c7*	*c5, c6, c7, c10, c13, c14*	0, 1, 2
Drain write buffer	*c7*	*c10*	4
Cache lockdown	*c9*	*c0*	0, 1
Round-robin replacement	*c15*	*c0*	0

Proteção de acesso aos recursos

- Alguns sistemas embarcados usam operações multitarefas e devem garantir que uma tarefa em execução não interrompa a operação de outras tarefas.
- A proteção dos recursos do sistema e outras tarefas contra acesso indesejado é chamada de **proteção**.
- Existem dois métodos para controlar o acesso aos recursos do sistema, **desprotegidos** e **protegidos**.
- Um sistema desprotegido depende apenas do software para proteger os recursos do sistema.
- Um sistema protegido depende de hardware e software para proteger os recursos do sistema.

Proteção de acesso aos recursos

- Sistema desprotegido
  - Conta apenas com proteção via software
  - Não há hardware dedicado à proteção
  - As tarefas devem cooperar entre si para garantir o acesso correto aos recursos
- Sistema protegido
  - Conta com proteção por software hardware
  - Hardware dedicado à proteção
  - As tarefas devem seguir certas regras determinadas pelo S.O. e pelo hardware, a fim de garantirem os privilégios de acesso necessários

Recursos

- Tipicamente, dois tipos de recurso precisam de proteção: Memória e periféricos de E/S
- No ARM, os periféricos são mapeados em memória
  - A proteção de acesso à memória também resulta na proteção aos dispositivos de E/S

MPU vs. MMU

- MPU (Memory Protection Unit)
  - Proteção de áreas (regiões) especificadas em software
- MMU (Memory Management Unit)
  - Proteção com uso de memória virtual

MPU – Regiões

- Uma região trata-se de um conjunto de atributos (configurados usando o CP15) associados a uma área de memória.
  - ID (de 0 a 7)
  - Permissões de acesso
  - Endereço inicial
  - Tamanho
  - Políticas de cache

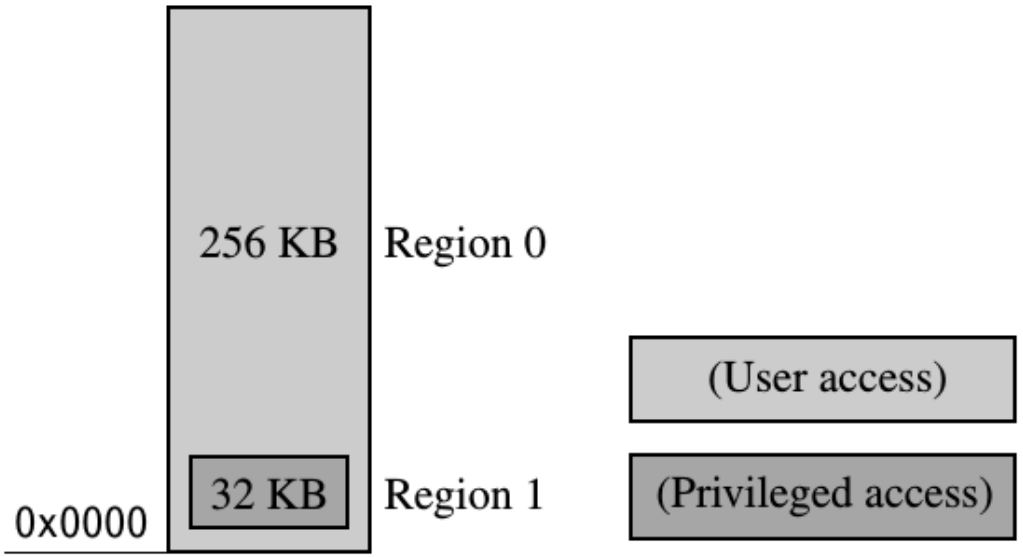
MPU – Regiões

- Algumas regras:
  - Regiões recebem um número que determina sua prioridade
  - Regiões podem sobrepor outras regiões
    - No caso de sobreposição, os atributos da região de maior prioridade tem precedência sobre as demais regiões. Essa prioridade só é aplicada aos endereços contidos na área sobreposta.
  - O tamanho da região é indicado em potências de 2, entre 4KB e 4GB
  - O endereço inicial de uma região deve ser múltiplo de seu tamanho.

MPU – Regiões

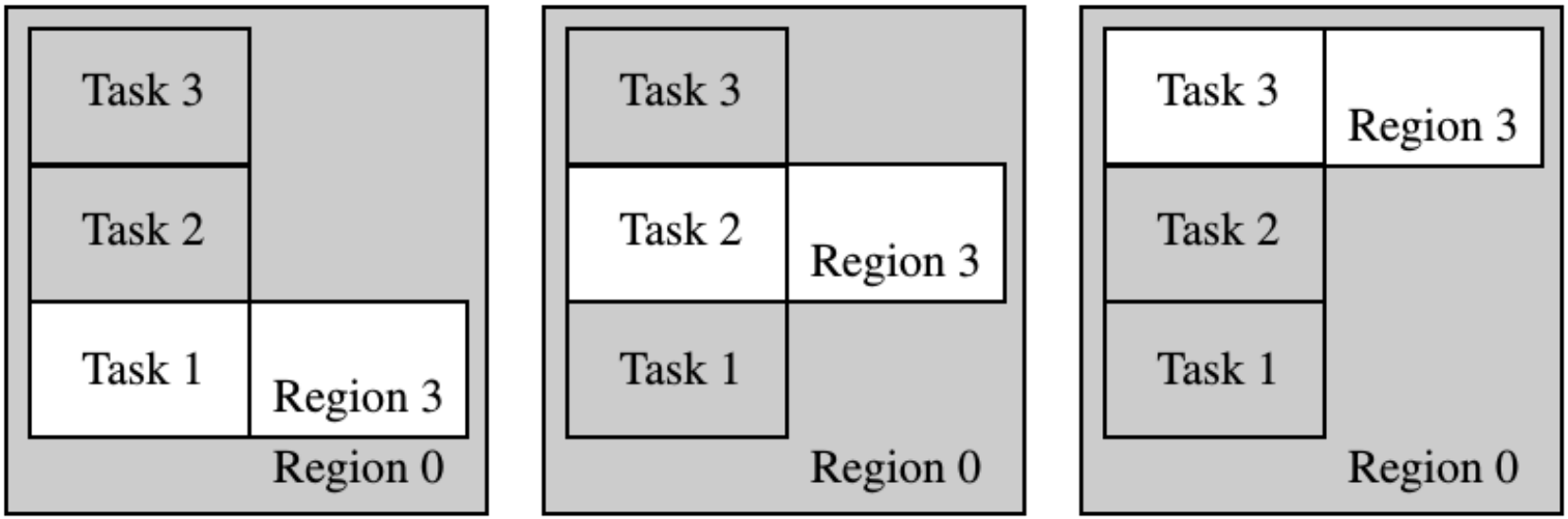
- Exceções de Prefetch Abort ou Data Abort
  - Quando o processador tenta acessar uma determinada região, a MPU **compara** as permissões de acesso com o modo de execução atual do processador. **Caso não haja permissão de acesso é gerada uma exceção**.
  - Ocorrem também ao tentar acessar um endereço de uma região não definida.

MPU – Regiões



ARM core	Number of regions	Separate instruction and data regions	Separate configuration of instruction and data regions
ARM740T	8	no	no
ARM940T	16	yes	yes
ARM946E-S	8	no	yes
ARM1026EJ-S	8	no	yes

Regiões em Background



MPU e o CP15

Coprocessor registers that control the MPU.

Function	Primary register	Secondary registers
System control	*c1*	*c0*
Region cache attributes	*c2*	*c0*
Region write buffer attributes	*c3*	*c0*
Region access permissions	*c5*	*c0*
Region size and location	*c6*	*c0* to *c7*

Definindo regiões (CP15:c6)

As etapas a seguir são necessárias para inicializar a MPU, os caches e o buffer de gravação:

1. Defina o tamanho e a localização das regiões de instruções e dados usando CP15:c6.
2. Defina a permissão de acesso para cada região usando CP15:c5.
3. Defina os atributos de cache e buffer de gravação para cada região usando CP15:c2 para cache e CP15:c3 para buffer de gravação.
4. Habilite os caches e o MPU usando CP15:c1.

Definindo regiões (CP15:c6)

CP15:c6 register format setting size and location of a region.

Field name	Bit fields	Comments
Base address	[31:12]	Address greater than 4 KB must be a multiple of the size represented in [5:1]
SBZ	[11:6]	Value "should be zero"
N	[5:1]	Size of region is 2^{N+1}, where $11 \leq N \leq 31$
E	[0]	Region enable, 1 = enable, 0 = disable

Definindo regiões

- **Exemplo** → Região 3, iniciando em 0x300000, com 256KB

```
MOV     r1, #0x300000              ; set starting address
ORR     r1, r1, #0x11<<1           ; set size to 256 KB
MCR     p15, 0, r1, c6, c3, 0
```

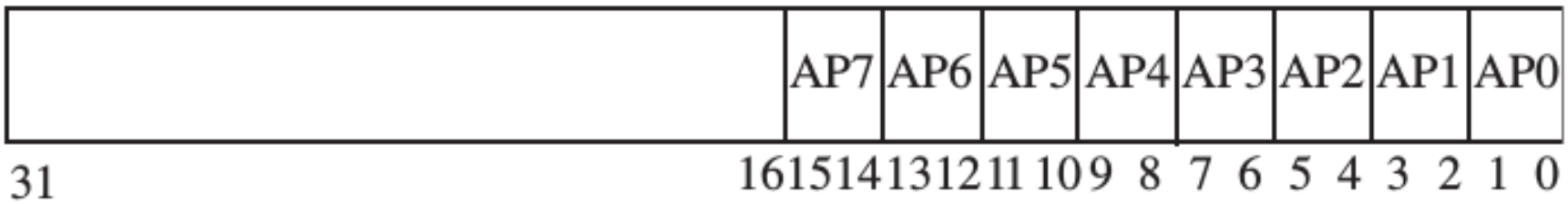
Definindo regiões

- **Exemplo** $\rightarrow$ Lendo tamanho e endereço da região 5, na arquitetura ARM940T

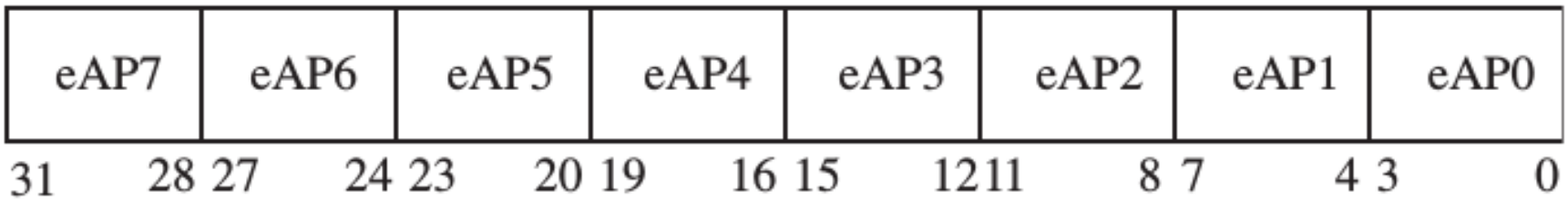
```
MRC     p15, 0, r2, c6, c5, 0  ; r2 = base/size Data Region 5
MRC     p15, 0, r3, c6, c5, 1  ; r3 = base/size Inst Region 5
```

Permissões de acesso (CP15:c5)

CP15:c5:c0 standard instruction region AP
CP15:c5:c1 standard data region AP



CP15:c5:c2 extended instruction region AP
CP15:c5:c3 extended data region AP



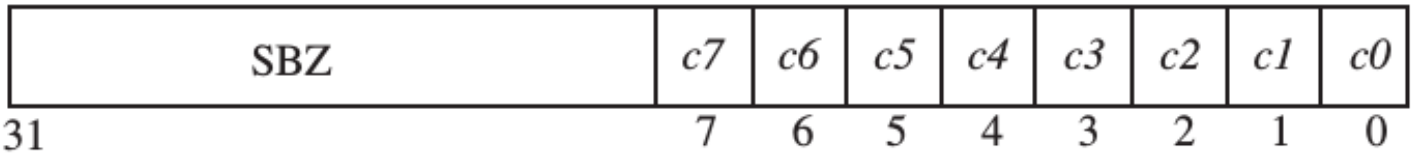
CP15 register 5 access permission register formats.

Permissões de acesso (CP15:c5)

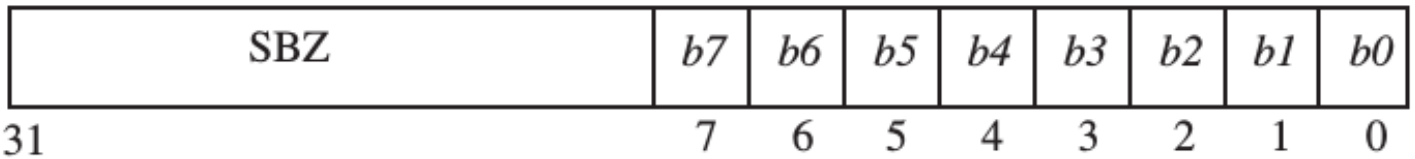
Supervisor	User	Standard AP value	Extended AP value
No access	no access	00	0000
Read/write	no access	01	0001
Read/write	read only	10	0010
Read/write	read/write	11	0011
Unpredictable	unpredictable	—	0100
Read only	no access	—	0101
Read only	read only	—	0110
Unpredictable	unpredictable	—	0111
Unpredictable	unpredictable	—	1000 to 1111

Configuração de cache e de write buffer (CP15:c2:c3)

CP15:c2:c0:0—D-cache
CP15:c2:c0:1—I-cache



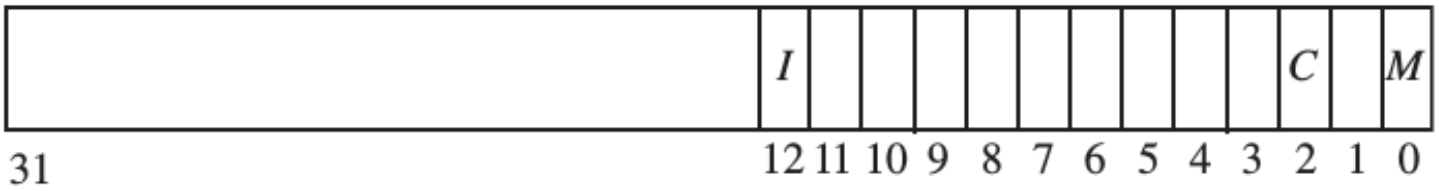
CP15:c3:c0:0—write buffer



SBZ = "should be zero"

CP15:c2 cache and CP15:c3 write buffer region registers.

Habilitando a MPU (CP15:c1)



Memory protection unit control bits in the CP15:c1:c0 control register.

Protection unit enable bits in CP15 control register 1.

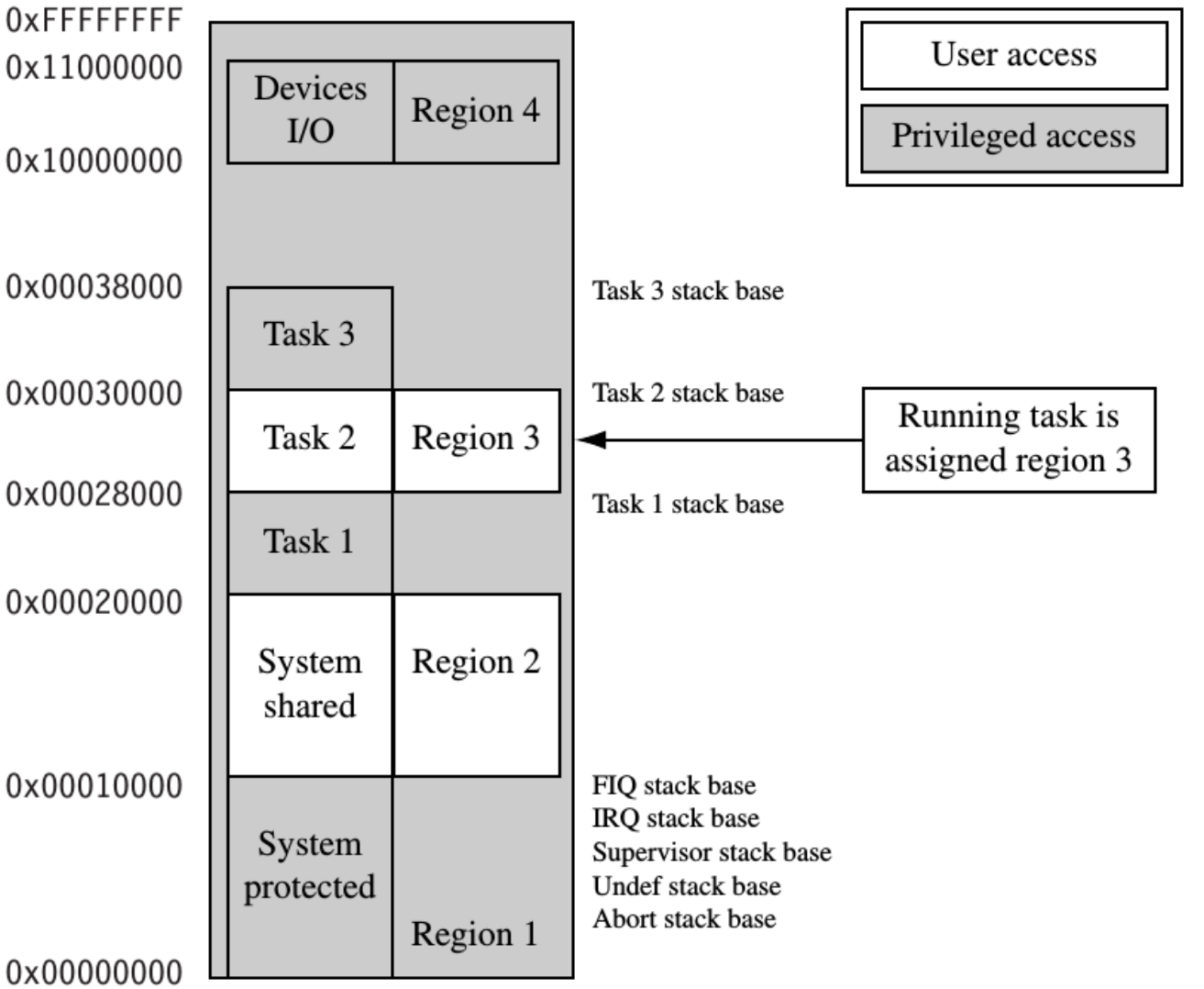
Bit	Function enabled	Value
0	MPU	0 = disabled, 1 = enabled
2	data cache	0 = disabled, 1 = enabled
12	instruction cache	0 = disabled, 1 = enabled

Exemplo (ver seção 13.3)

Table 13.10 Memory map of example protection system.

Function	Access level	Starting address	Size	Region
Protect memory-mapped peripheral devices	system	0x10000000	2 MB	4
Protected system	system	0x00000000	4 GB	1
Shared system	user	0x00010000	64 KB	2
User task 1	user	0x00020000	32 KB	3
User task 2	user	0x00028000	32 KB	3
User task 3	user	0x00030000	32 KB	3

Exemplo (ver seção 13.3)



QXD0133 - Arquitetura e Organização de Computadores II

Universidade Federal do Ceará - Campus Quixadá

Thiago Werlley
thiagowerlley@ufc.br

18 de outubro de 2025

Capítulo 13