ARM v7-M architecture

R. Ferrero

Politecnico di Torino

Dipartimento di Automatica e Informatica (DAUIN)

Torino - Italy

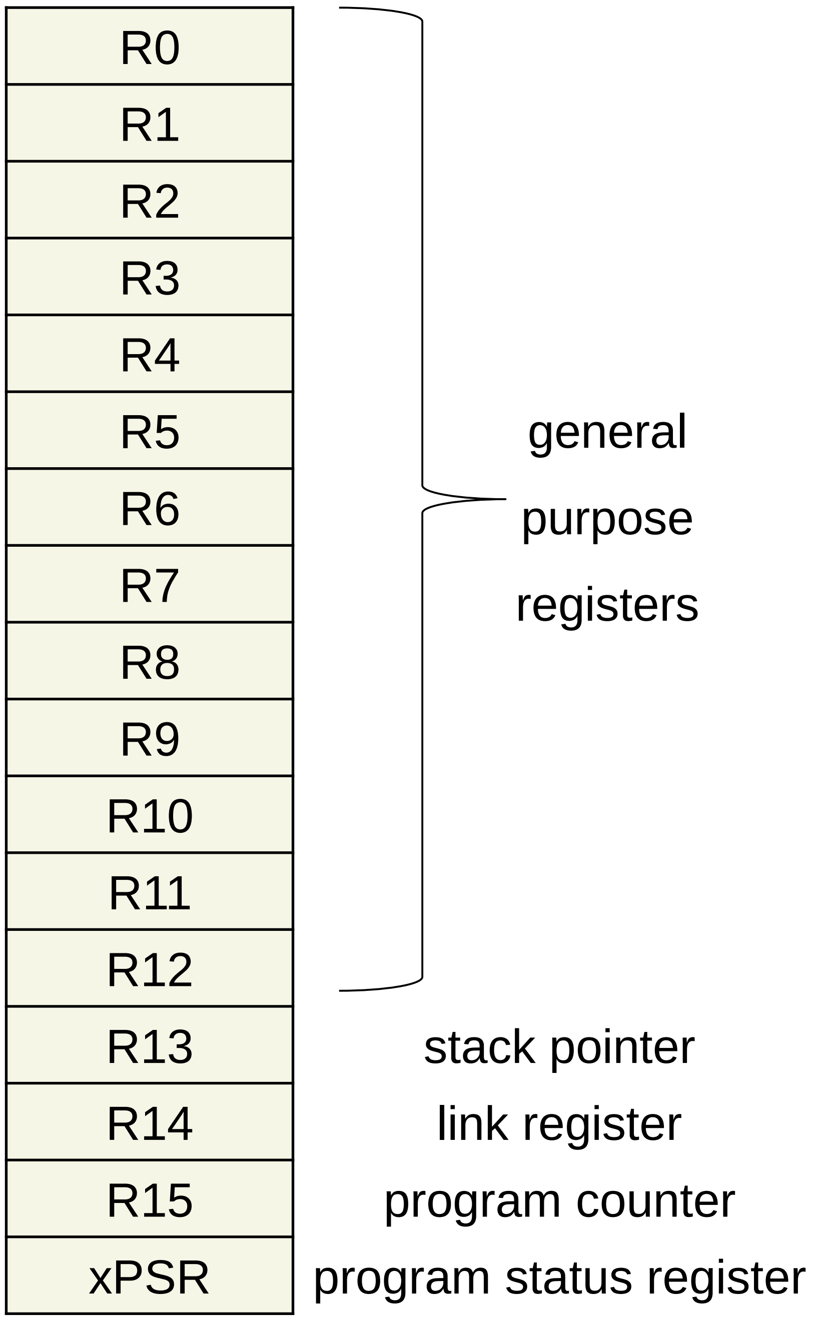
Data types

- The following data types are supported:
  - byte: 8 bits
  - halfword: 16 bits
  - word: 32 bits
- Registers are 32-bit wide.
- Instructions are 16 or 32 bits long.

不同于8086，8086中，word 16 bits
并且没有halfword这个类型

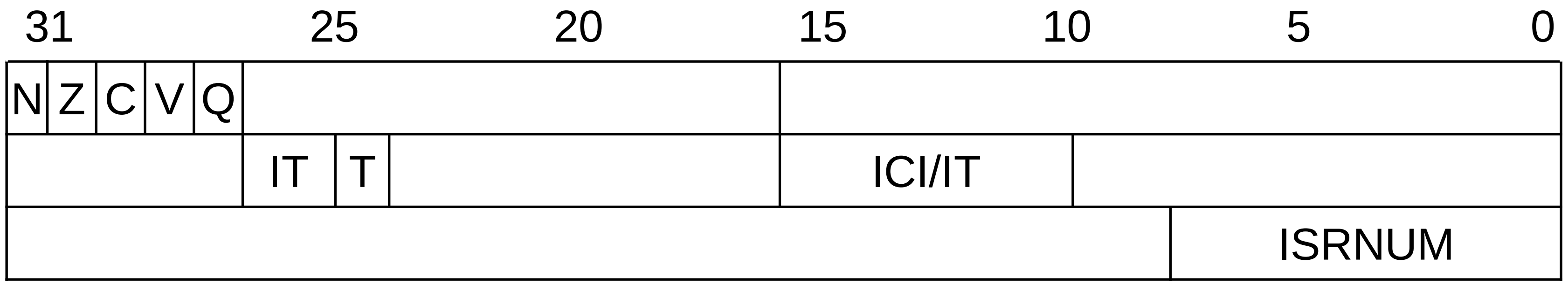
Registers

- Registers are used to store temporary information.



Program Status Register

- It can be accessed all at once or as a combination of 3 registers:
  - Application Program Status Register (APSR)
  - Execution Program Status Register (EPSR)
  - Interrupt Program Status Register (IPSR)



Application Program Status Register

- It contains:
  - N: Negative result from ALU
  - Z: Zero result from ALU
  - C: ALU operation Carried out
  - V: ALU operation oVerflowed
  - Q: “sticky” flag for saturation arithmetic.

EPSR and IPSR

- The Execution Program Status Register contains:
  - IT: IF-THEN instruction status bits
  - ICI: Interrupt-Continuable Instruction bits
  - T: Thumb bit
- The Interrupt Program Status Register contains an exception number used in exception handling.

Processor modes

- Two operating modes:
  - thread mode: on reset or after an exception
  - handler mode: when an exception occurs
- Two access levels:
  - user level: limited access to resources
  - privileged level: access to all resources
- Handler mode is always privileged.

Vector table

- Each entry manages an exception, interrupt, or other atypical event such as a reset.
- The vector table contains either ARM instructions or addresses in memory.
- In the v7-M architecture, there are addresses.

Vector table in v7-M architecture

Exception Type	Position	Vector Address
(Top of Stack)	0	0x00000000
Reset	1	0x00000004
NMI	2	0x00000008
Hard fault	3	0x0000000C
Memory management fault	4	0x00000010
Bus fault	5	0x00000014
Usage fault	6	0x00000018
SVcall	11	0x0000002C
Debug monitor	12	0x00000030
PendSV	14	0x00000038
SysTick	15	0x0000003C
Interrupts	≥16	≥0x00000040

Features of ARM Instruction Sets

- Most instructions execute in a single cycle.
- Instructions can be conditionally executed.
- A load/store architecture
  - Data processing instructions act only on registers
  - Two or three operand formats
  - Combined ALU and shifter
  - Memory access instructions with auto-indexing.

Instruction length

- There are 3 kinds of instructions:
  - ARM instructions: 32 bits wide
  - Thumb instructions: 16 bits wide
  - Thumb-2 instructions: 32 bits wide
- Example: sum of registers
  - 32 bits: `ADD r0, r0, r2`
  - 16 bits: `ADD r0, r2`
- ARM v7-M architecture supports Thumb and Thumb-2 instructions only.

Example: shifting data

```
    AREA     |.text|, CODE, READONLY
Reset_Handler   PROC
    EXPORT  Reset_Handler [WEAK]
    MOV r0, #0x11 ;load initial value
    LSL r1, r0, #1 ;shift 1 bit left
    LSL r2, r1, #1 ;shift 1 bit left
stop B stop            ;stop program
    ENDP
```

Example: factorial calculation

```
    AREA    |.text|, CODE, READONLY
Reset_Handler   PROC
    EXPORT  Reset_Handler [WEAK]
    MOV r6, #10 ;load 10 into r6
    MOV r7, #1
loop CMP r6, #0
    ITTT GT    ;start of IF-THEN block
    MULGT r7, r6, r7
    SUBGT r6, r6, #1
    BGT loop   ;end of IF-THEN block
stop B stop          ;stop program
    ENDP
```

Example: swapping registers

```
    AREA    |.text|, CODE, READONLY
Reset_Handler   PROC
    EXPORT  Reset_Handler [WEAK]
    LDR r0,=0xF631024C ;load some data
    LDR r1,=0x17539ABD ;load some data
    EOR r0, r0, r1     ;r0 XOR r1
    EOR r1, r0, r1     ;r1 XOR r0
    EOR r0, r0, r1     ;r0 XOR r1
stop B stop            ;stop program
    ENDP
```