ITMC301
L4
ARM Modes

By: Dr. Abdussalam Nuri Baryun

Mobile Computing Department, Faculty of Information Technology, University of Tripoli

ARM Rigisters

- There are up to 18 active registers: 16 data registers and 2 processor status registers.
- The data registers are visible to the programmer as *r0 to r15.*
- The ARM processor has three registers assigned to a particular task or special function: *r13, r14, and r15. They are frequently given different labels to differentiate them from the* other registers.
- In addition to the 16 data registers, there are two program status registers: *cpsr and spsr* (the current and saved program status registers, respectively).

Generic Program Status Register (GPSR)

- The current processor mode is stored in the *cpsr*. It holds the current status of the processor core as well interrupt masks, condition flags, and state. The state determines which instruction set is being executed.

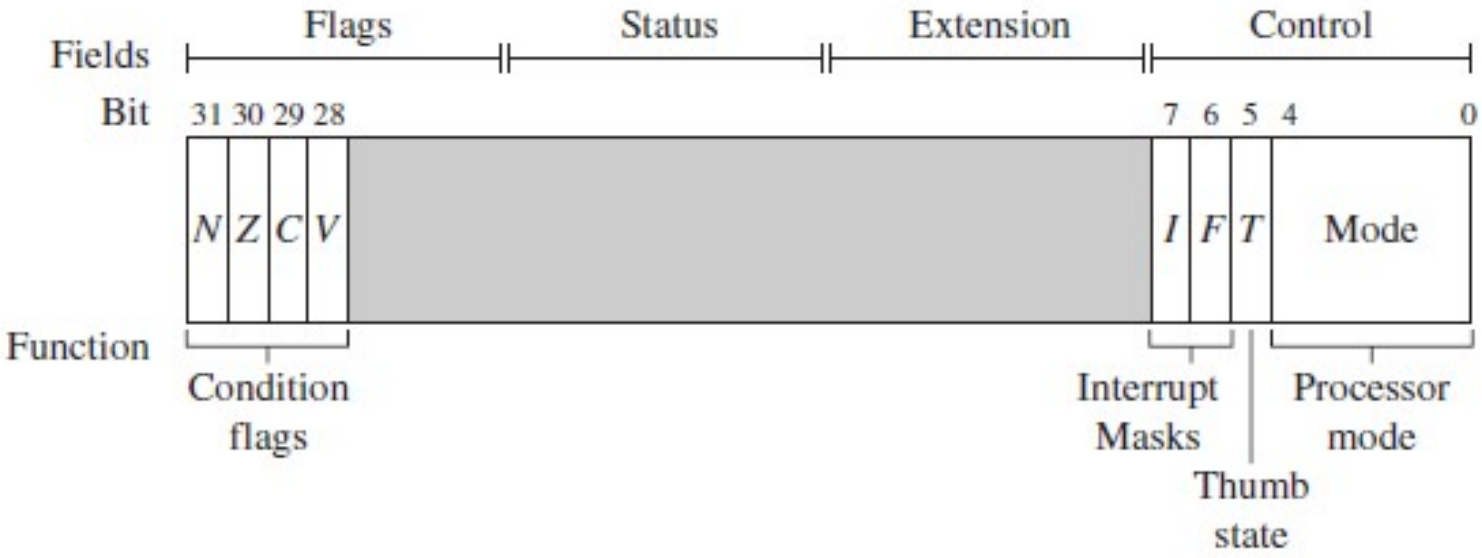


Figure 2.3 A generic program status register

- Register *r13 is traditionally used as the stack pointer (sp) and stores the head of the stack* in the current processor mode.
- Register *r14 is called the link register (lr) and is where the core puts the return address* whenever it calls a subroutine.
- Register *r15 is the program counter (pc) and contains the address of the next instruction* to be fetched by the processor.

Banked Registers

- *Banked registers are* 20 registers hidden from a program at different times.
- *In Figure 2.4 they are identified* by the shading in the diagram. They are available only when the processor is in a particular mode.

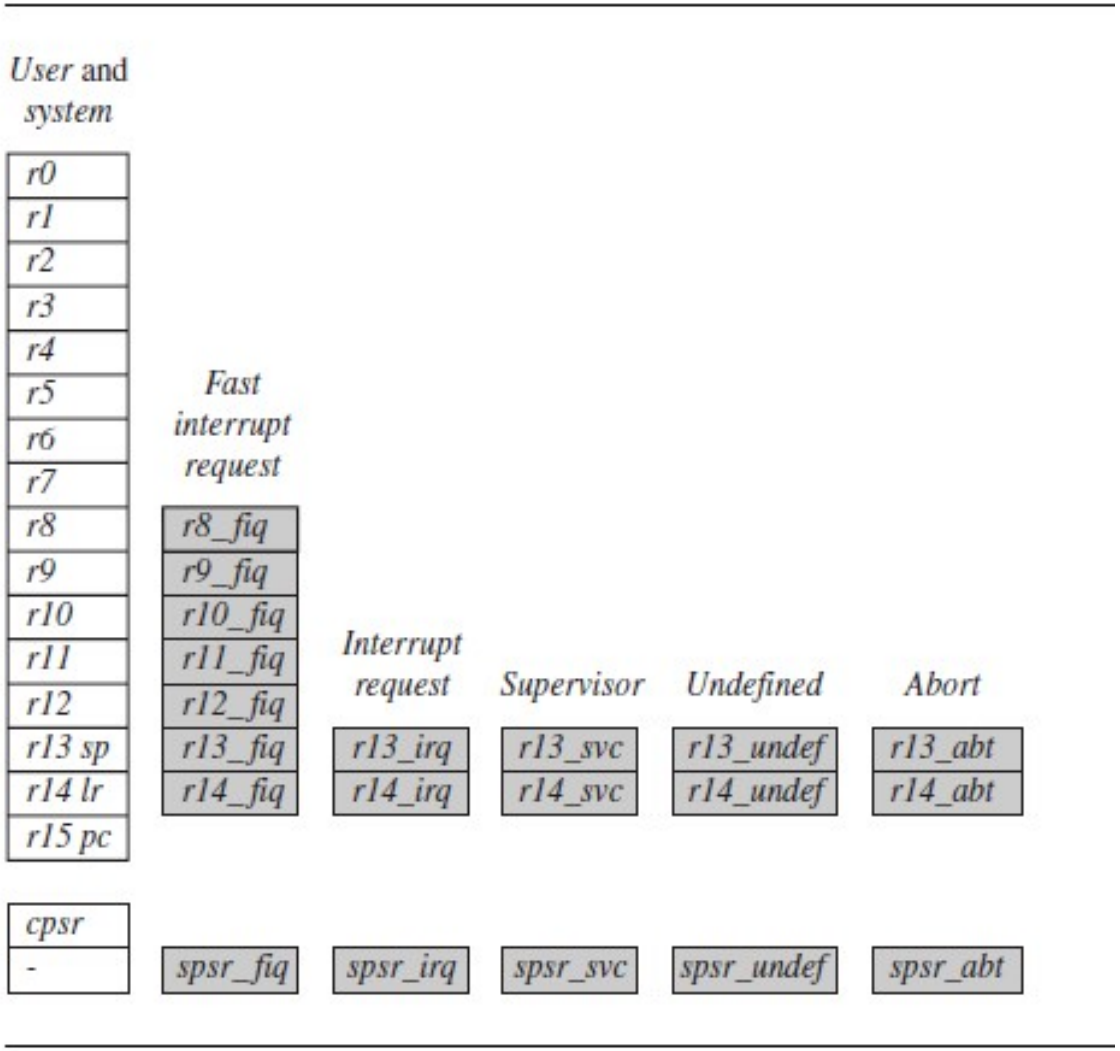


Figure 2.4 Complete ARM register set.

- The Programmers Model can be split into following elements
  - Data types
  - processor modes
  - processor registers
- The typical application will run in an unprivileged mode know as “User” mode, whereas the various exception types will be dealt with in one of the privileged modes : Fast Interrupt, Supervisor, Abort, Normal Interrupt and Undefined.

Modes

- **The ARM has seven basic operating modes:**
  - **User** : unprivileged mode under which most tasks run (normal program execution mode).
  - **FIQ** : entered when a high priority (fast) interrupt is raised (supports a high-speed data transfer or channel process)
  - **IRQ** : entered when a low priority (normal) interrupt is raised (general-purpose interrupt handling).
  - **Supervisor** : entered on reset and when a Software Interrupt instruction is executed (used for operating system)
  - **Abort** : used to handle memory access violations (implements virtual memory and/or memory protection).
  - **Undef** : used to handle undefined instructions (supports software emulation of hardware coprocessors).
  - **System** : run privileged operating systems tasks (ARMv4 and above)

Processor mode		Mode number	Description
User	usr	0b10000	Normal program execution mode
FIQ	fiq	0b10001	Supports a high-speed data transfer or channel process
IRQ	irq	0b10010	Used for general-purpose interrupt handling
Supervisor	svc	0b10011	A protected mode for the operating system
Abort	abt	0b10111	Implements virtual memory and/or memory protection
Undefined	und	0b11011	Supports software emulation of hardware coprocessors
System	sys	0b11111	Runs privileged operating system tasks (ARMv4 and above)

Mode changes can be made under software control, or can be caused by external interrupts or exception processing.

Most application programs execute in User mode. When the processor is in User mode, the program being executed is unable to access some protected system resources or to change mode, other than by causing an exception to occur

The modes other than User mode are known as *privileged modes*. They have full access to system resources and can change mode freely. Five of them are known as *exception modes*:

- FIQ
- IRQ
- Supervisor
- Abort
- Undefined.

These are entered when specific exceptions occur. Each of them has some additional registers to avoid corrupting User mode state when the exception occurs

The remaining mode is System mode, which is not entered by any exception and has exactly the same registers available as User mode. However, it is a privileged mode and is therefore not subject to the User mode restrictions. It is intended for use by operating system tasks that need access to system resources, but wish to avoid using the additional registers associated with the exception modes. Avoiding such use ensures that the task state is not corrupted by the occurrence of any exception.

The Registers

- **ARM has 37 registers all of which are 32-bits long.**
  - 1 dedicated program counter
  - 1 dedicated current program status register
  - 5 dedicated saved program status registers
  - 30 general purpose registers

- **The current processor mode governs which of several banks is accessible. Each mode can access**
  - a particular set of r0-r12 registers
  - a particular r13 (the stack pointer, sp) and r14 (the link register, lr)
  - the program counter, r15 (pc)
  - the current program status register, cpsr

  Privileged modes (except System) can also access
  - a particular spsr (saved program status register)

The Registers

The ARM registers are arranged into several banks, with the accessible bank being governed by the current processor mode. The core can access:

A particular set of 13 general purpose registers (r0 - r12).

A particular r13 - which is typically used as a stack pointer. This will be a different r13 for each mode, so allowing each exception type to have its own stack.

A particular r14 - which is used as a link (or return address) register. Again this will be a different r14 for each mode.

r15 - whose only use is as the Program counter.

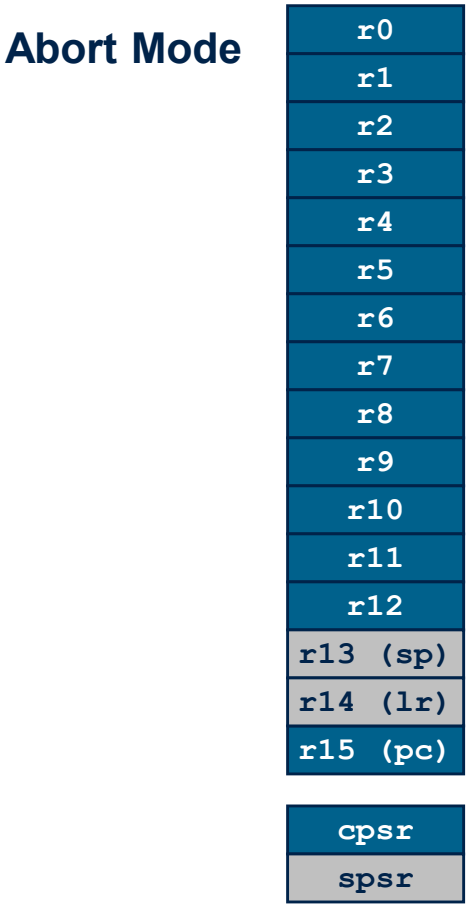
The CPSR (Current Program Status Register) - this stores additional information about the state of the processor:

And finally in privileged modes, a particular SPSR (Saved Program Status Register). This stores a copy of the previous CPSR value when an exception occurs. This combined with the link register allows exceptions to return without corrupting processor state.

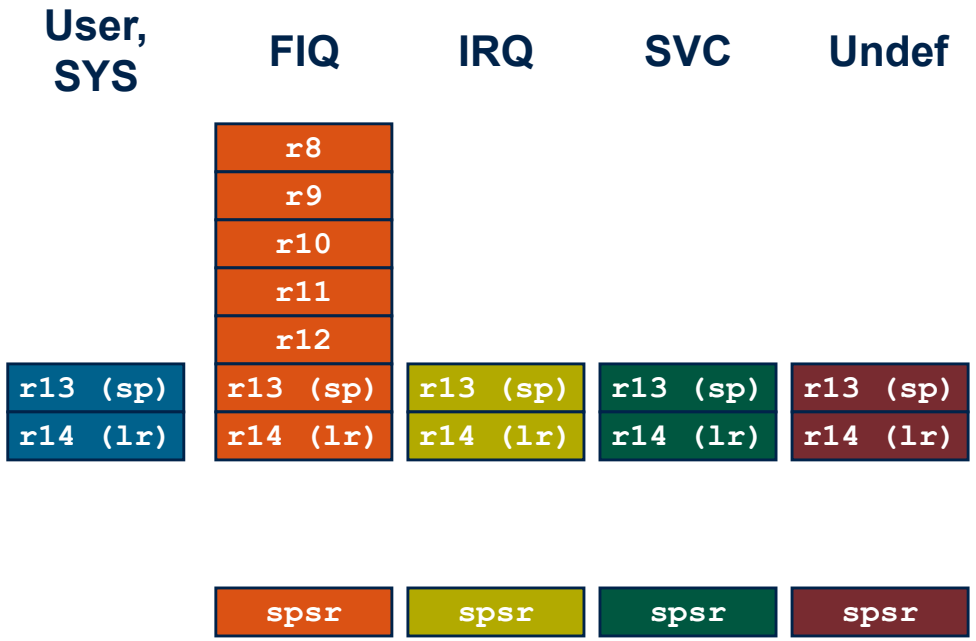
Special Registers

- **Special function registers:**
  - **PC** (R15): Program Counter. Any instruction with PC as its destination register is a program branch
  - **LR** (R14): Link Register. Saves a copy of PC when executing the BL instruction (subroutine call) or when jumping to an exception or interrupt routine
    - It is copied back to PC on the return from those routines
  - **SP** (R13): Stack Pointer. There is **no stack** in the ARM architecture. Even so, R13 is usually reserved as a pointer for the program-managed stack
  - **CPSR** : Current Program Status Register. Holds the visible status register
  - **SPSR** : Saved Program Status Register. Holds a copy of the previous status register while executing exception or interrupt routines
    - It is copied back to CPSR on the return from the exception or interrupt
    - No SPSR available in User or System modes

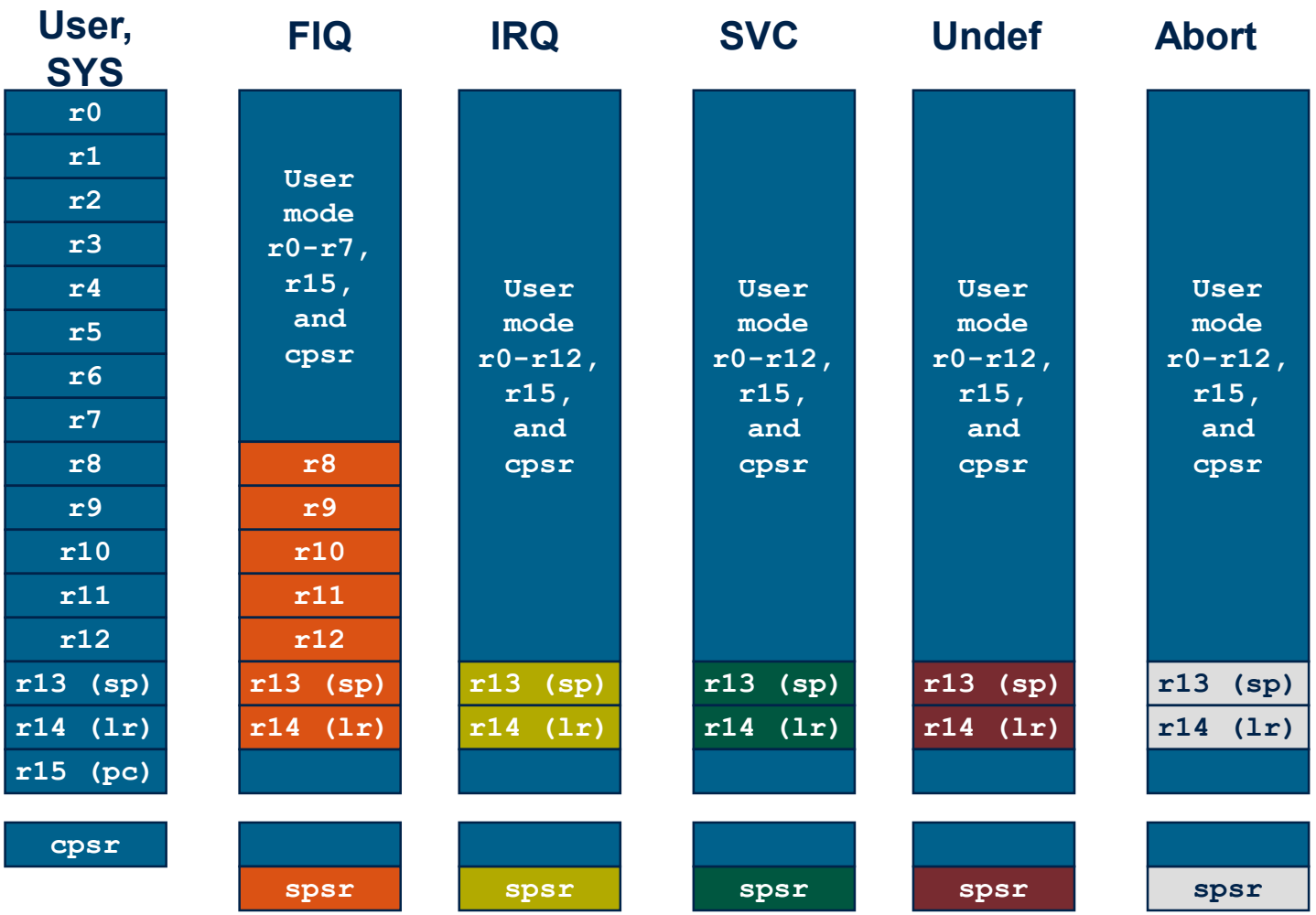
Current Visible Registers



Banked out Registers

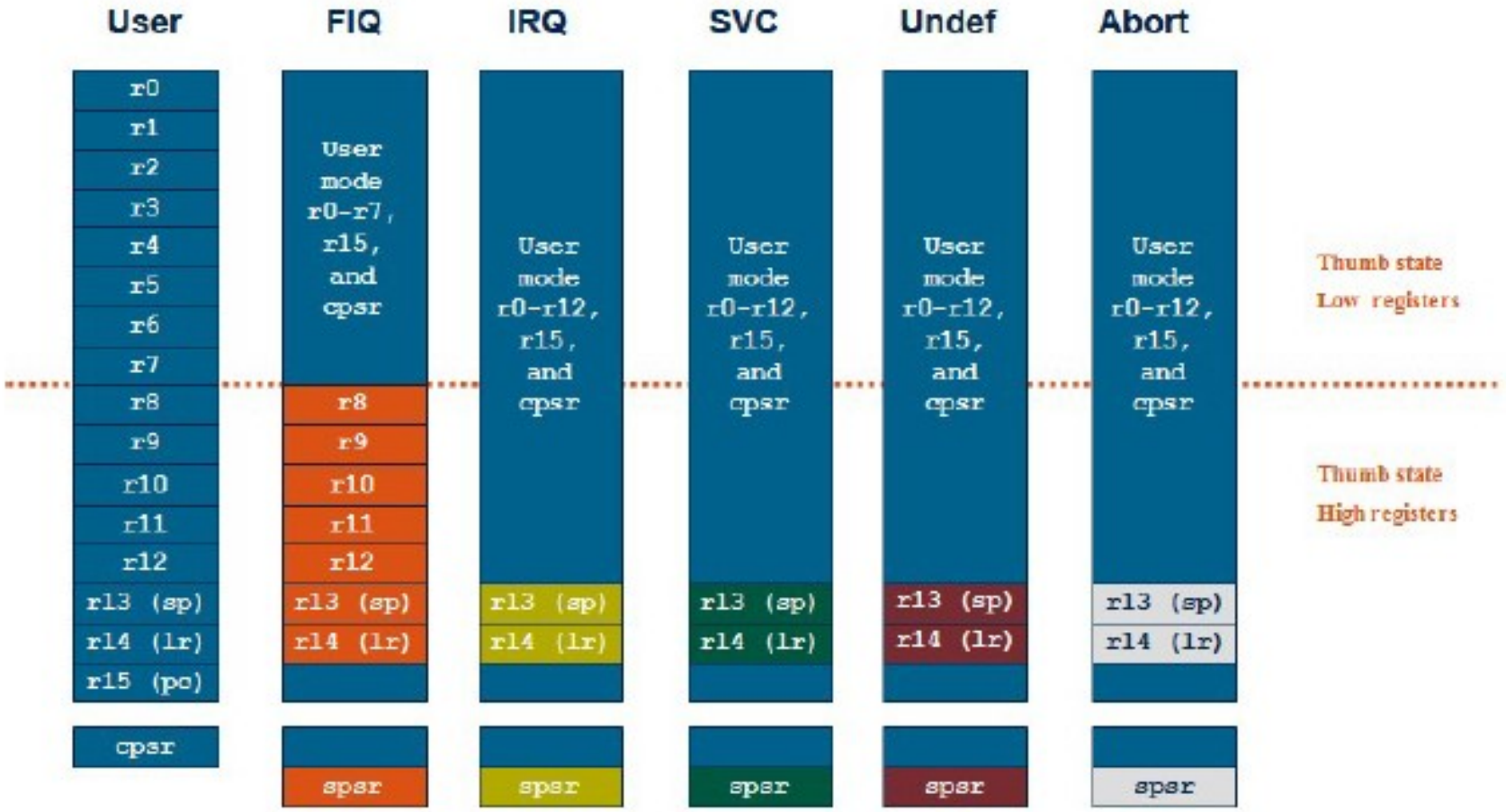


Register Organised



Note: System mode uses the User mode register set

Register Organise



Note: System mode uses the User mode register set