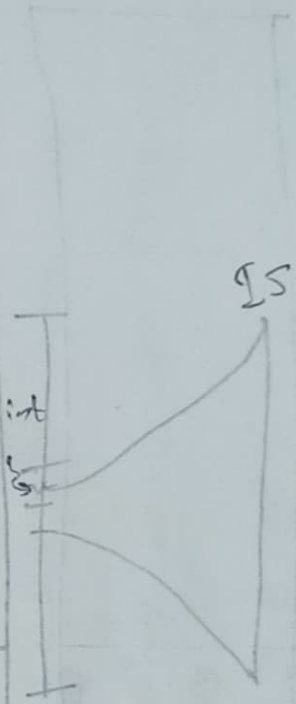
ARM7 Registers (Programmer Model)



There are 37 registers in ARM7.

USR/SYS	FIQ	IRQ	SVR	ABT	UND
R0	R0	R0	R0	R0	R0
⋮	⋮	⋮	⋮	⋮	⋮
R7	R7	R7	R7	R7	R7
R8	R8 (5 New REGs of FIQ)	R8	R8	R8	R8
⋮	⋮	⋮	⋮	⋮	⋮
R12	R12	R12	R12	R12	R12
R13 - SP	R13 - SP	R13 - SP	R13 - SP	R13 - SP	R13 - SP
R14 - LR	R14 - LR	R14 - LR	R14 - LR	R14 - LR	R14 - LR
R15 - PC	R15 - PC	R15 - PC	R15 - PC	R15 - PC	R15 - PC
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR	SPSR	SPSR	SPSR	SPSR

R0 – R15 → 16
CPSR → 1
} 17

5 New SP → 5
5 New LR → 5
5 New SPSR → 5
5 New FIQ → 5
} 20

⇒ 37 regs

Lec 05
14 August, 2021

USR/SYS → 17

* There are 7 modes. 90% of its lifetime an ARM processor spends time in user (USR) mode. They have some registers.

* R0 to R15 are general purpose registers. But some of them have special use.

R13 → SP (stack pointer). it contains the address of top of stack.

R14 → LR (link register)

R15 → PC (program pointer) : it contains the address of next instruction.

CPSR → it contains the flags of ARM. (flag register)

there are R0 to 15 => 16 Reg

CPSR => 1 Reg

Total 17 registers are used in USER mode.

out of which there are more 20 regs (physically new registers but with same name) in di fer different Modes.

User mode and System mode use the same set of registers.

Because System Mode is not different from user mode but they are the privileged part from the user. There does not need any interrupt to enter the system mode from user mode to invoke system mode.
So these two modes use same set of registers.

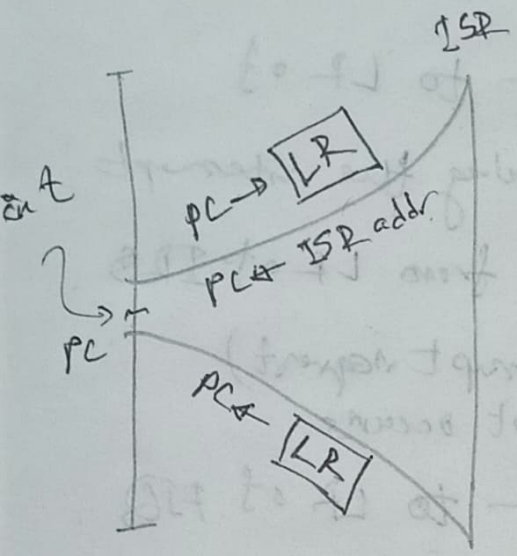
LR → 5

User mode and System mode do not have need any interrupt. they are auto invoked.

The other five modes are invoked by different seven interrupts.

In other processors when any interrupt occurs we go to the ISR and after operation we return to the main program.

But in ARM7 when an interrupt occurs, not only we go to the ISR but also the mode changes.



In normal process, before going to ISR, PC stores it address to stack and then receive ISR address. After finishing ISR operation. PC gets back its address which is stored before going to ISR.

In ARM stack is not used. because stack is in memory. Memory operation is slow also it disturbs pipelining.

So instead of using stack

ARM uses (LR link Register)

LR stores the address of previous modes.

When any interrupt occurs, Let IRQ (Interrupt Request)

PC stores its value to LR of IRQ. after finishing the interrupt PC gets its value from. LR of IRQ.

when FIQ (fast interrupt request) interrupt occurs

PC stores its value to LR of FIQ.

Also when going back from this interrupt PC gets its value from LR of FIQ.

Similarly for five modes there are five LR registers.

LR

The register in user mode does not contain any address of other modes. Because users always stays in user mode. So users mode does need to go back anywhere.

But this LR is used when we use any function. When any function is called the value in pc is stored in LR of user mode.

SPSR → S

When we are in User mode, there are occuring operation and they may have some flag values. The flag values are stored in CPSR.

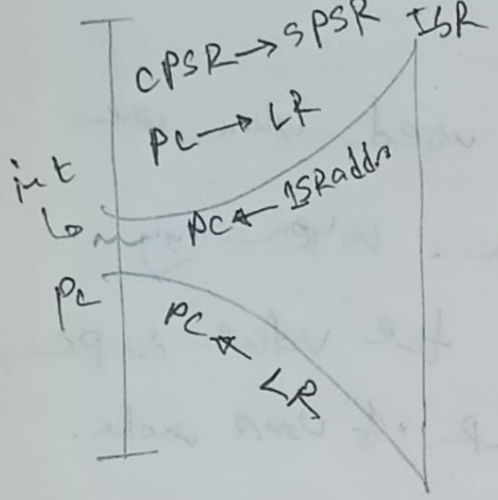
At that time if any interrupt is called, the CPSR will be used for LSR of that interupt.

When back to main program

to usr mode the flags values in previous operation will be lost for going to ISR.

So CPSR values should be stored in any register called SPSR.

So for five modes, there are five SPSR.



There is no SPSR in USER/SYS Mode. Because user mode does need to go back to any other mode.

SP → S

In user mode we may have use some stack operation. at that time we may go to any other mode. There we may use stack operation.

while coming to user mode.

the top of stack values might be different.

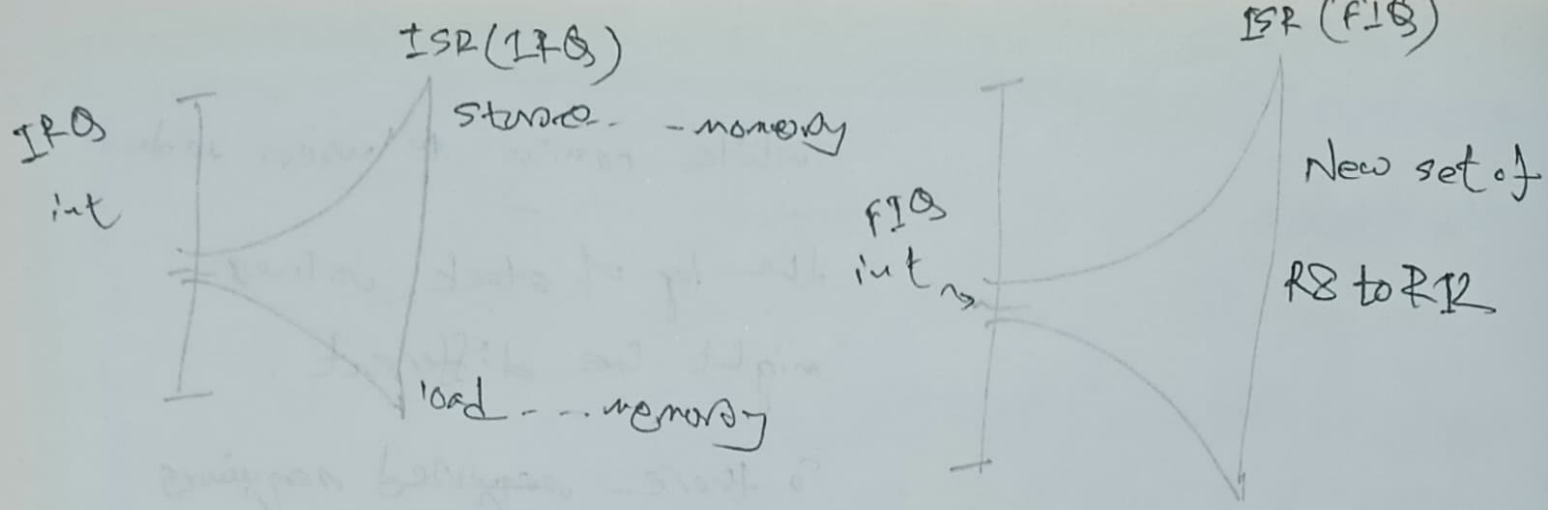
So there required requires 5 ~~regist~~ registers for each of the 5 different modes.

Every modes needs its own ~~SP~~ SP registers.

For more five modes there are need 5 SP registers.

FIQ regs → 5

In normal interrupt, before going to ISR the values of registers (general purpose registers) are stored in memory. Because when back to user mode we get the previous value.



In fast interrupt (FIQ)
we use a new set of
registers (R8 to R12 => 5 regs).
By using these new regs
we do not need to load and store
with memory which helps
~~in~~ save time
and also in pipelining.
Thus the general purpose
register values do not get
changed while using FIQ.

In normal situation we use IRQ
interrupt, in critical situation
we use FIQ interrupt.