ARM (Embedded C)

ARM :- Advanced RISC Machine

* Contents of ARM :

1) Fundamentals (microprocessor (μp), microcontroller (μC), core, CISC, RISC).
2) Features of LPC2129, Pinning info.
3) Assembly basics.
4) Timer (delay)
5) GIPO Programming
6) UART (Bluetooth, wifi)
7) SPI (memory card)
8) ADC (Sensor)
9) I2C (time)
10) CAN (automative protocol)
11) Interrupt Handling.

ARM is 32 bit

Protocol means exchange the data.

Application } User space
C, C++
Python, Java
.net, linux

Strong C - fan, Device drivers
Strong linux - / Kernel space } Kernel Space
Prog., strong emb. C ↳ module

Motherboard

* Microcontroller

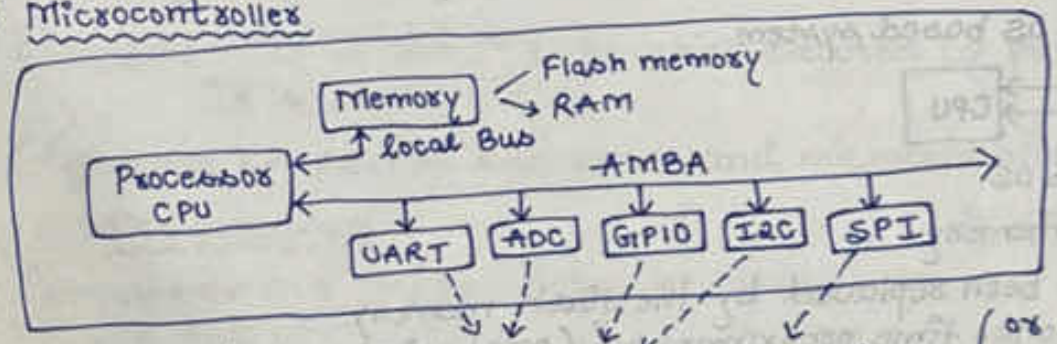


Fig :- microcontroller (ARM)

on-chip peripherals of μC (or device controllers)

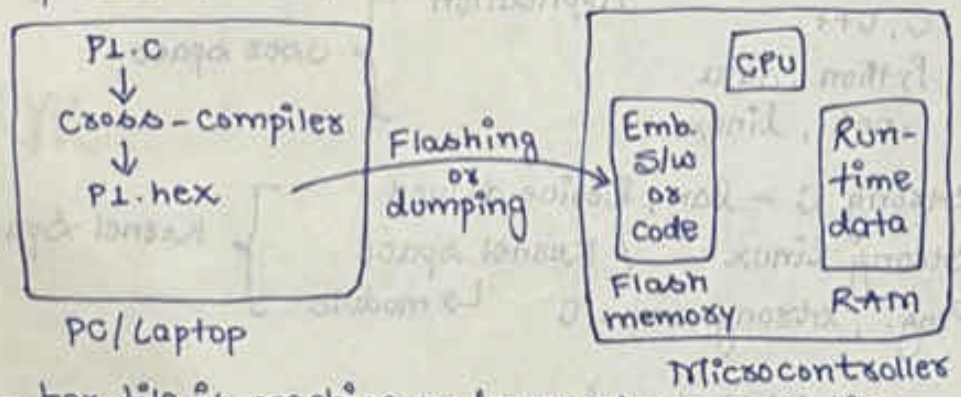
- In ARM controller, memory is interfaced to the processor using local bus.
- In ARM controller, on-chip peripherals of μC is interface to the processor using Bus (AMBA).

* List of microcontroller families :-

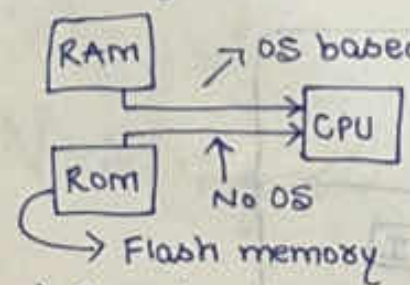
1) 8051 family
2) Advance versatile RISC (AVR) family. } slow in process as
3) Peripheral Interface Controller (PIC) family } compared to ARM
4) Advance RISC machine (ARM) family.

* Memory :- 1) Flash memory 2) RAM.
* Flash memory and RAM memory is mandatory in every microcontroller.
* EEPROM - Electrical erasable programming read only memory.
EEPROM → Permanent Read/write data memory
* Flash memory cannot store the data because in flash memory is used to store embedded software (program).
* In RAM memory, if you stop the power, it erase the data.
* EEPROM → Preserved the data permanently. (Run time / older data).
* RAM is use to store the runtime data.

* What is microcontroller ?
⇒ It is a programmable chip which contains in-built processor, memory and peripherals like GPIO, UART, timer ----



P1.hex - hex file is machine understandable code file.



Rom - 1) Rom has been replaced by the flash memory
2) Rom is one-time programmable (OTP) chip.
3) Rom is used to store programmable software in older microcontrollers.

* Rom is allowed to dump the code only one time.
* Flash memory is re-programmable memory. It is used to store embedded program and embedded software in new microcontrollers. It is permanent Read / writeable programmable.

It is permanent Read / writeable programming.
Local variable and global variable is store in RAM and local variable & global variable data is run-time data.

Flash memory is called as programmable or code memory.

* Code write → editor
  Code compile → compiler
  Code is executed by processor / CPU.

1) The role of processor / CPU is to execute embedded code and embedded software.
2) Processor can execute only one instruction at a time.
   Every instruction has different code (unique number) called as opcodes and processor is deal with only number (opcode).

In flash memory, only opcodes or number is there.

C - Program		Assembly Code		opcodes
↓				
main()	→	ASSM inst 1	→	num 1
{	→	ASSM inst 2	→	num 2
- - -	→	ASSM inst 3	→	num 3
- - -	→	ASSM inst 4	→	num 4
- - -				
}				

* Embedded software code always in infinite loop.

* Boot loader :-
1) Boot loader is used to store the code (or receive the code) in flash memory.
2) Boot loader is receiving .hex file from computer and store it in flash memory.

* Bias Code :- It is the 1st code which executed by processor.
  It is store in Rom.

* Difference between microprocessor and microcontroller

Microprocessor	Microcontroller
1. Processor doesn't contains internal memory and peripherals	1. Microcontroller contains internal memory and peripherals (like UART, timer, GPIO ----).
2. Processors are designed for desktop, PC's and laptops.	2. Microcontroller are designed for embedded system.
3. Power consumption is more.	3. Power consumption is less. (Heat dissipation is also less).
4. Processing is faster compared to the processor given inside µC. eg:- Intel, i3, i5, AMD....	4. Processing is slower compared to processors. eg:- 8051, AVR, PIC and ARM based microcontrollers. (Cost is less)

* Embedded System :-

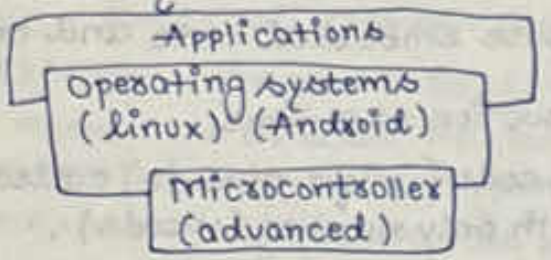
There are two types of embedded system :-

1. Embedded system with OS
2. Embedded system without OS

Intel / AMD runs 32768 program at a time.

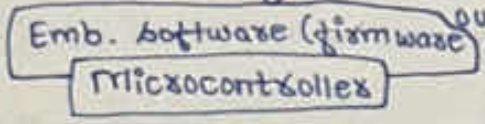
App termux → linux RTOS - Real time operating system

1) Embedded system with OS :-



ex :- Smartwatch, smart TV, wifi, Routes, N/w switch, smart-mobile.

2) Embedded system without OS :-



ex :- AC, washing machine, remote, bluetooth fan, bluetooth headphones.

* Backend of Android, Ubuntu is linux.
* Advanced microcontroller includes higher frequency, MPU (memory protection unit), MMU (memory management unit), TPU (Tensor processing unit).
* All these unit are missing in microcontroller of embedded system without OS.
* RTOS is embedded software. RTOS gives multidisection of environment.

Compilation stage in C → P1.c → cc P1.c → a.out

compilation stages in emb. → P1.c --- (C code) → Assembly code → opcode → executable code

Microprocessor		data bus	address bus
designed by intel	8085 (8 bits)	8 bits	16 bits
	8086 (8 bits)	16 bits	20 bits
	8088 (8 bits)	8 bits	20 bits

(8) bits microprocessor → maximum 2^8 instructions.

* If 32 bit processor, then processor can do 32 arithmetic logical operations in 1 clock cycle.
* If 16 bit processor, then processor can do 16 (ALU) operation in 1 clock

* If the processor is 8 bits, then ALU processing capacity could be 8 bits.

Instruction	opcode
ADD	10
SUB	20
MUL	30
DIV	40
MOV	50

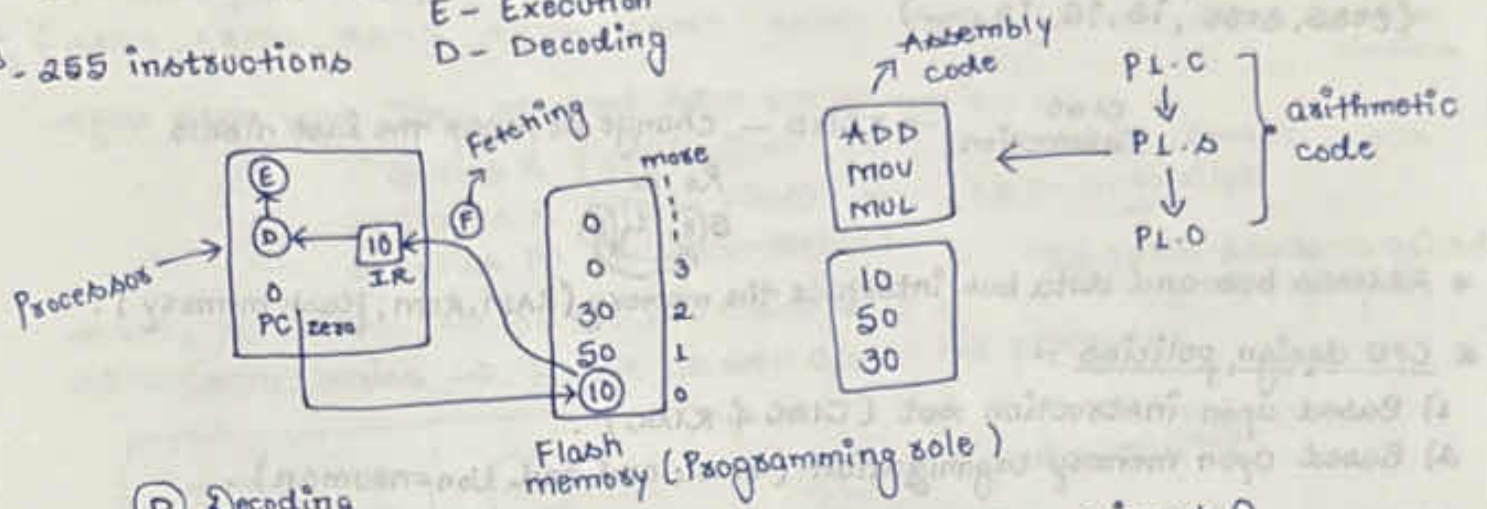
} assumed opcode.

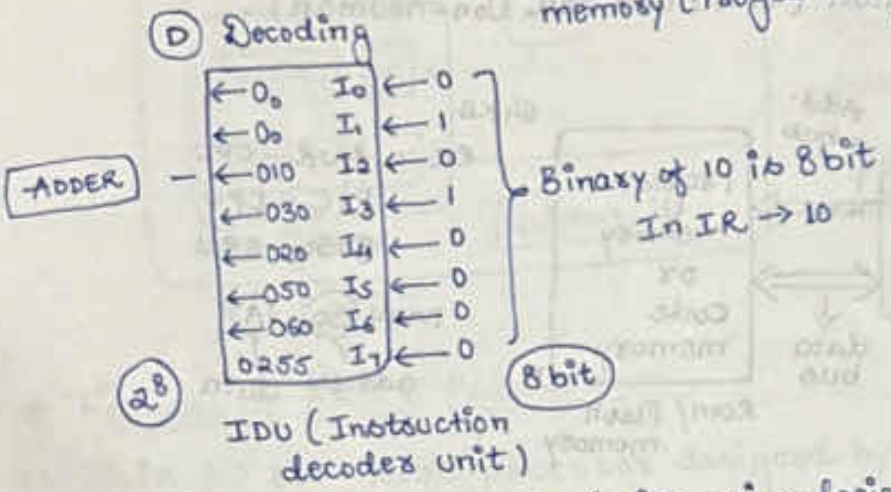
Hexe. Program counter is zero '0'.

PC - Program Counter
IR - Instruction register
E - Execution
D - Decoding

IDU - Instruction Decoder Unit

2^8 - 255 instructions





I_0 → input 0
O_0 → output 0

* Processor never stops execution.

* ADDER, MOV, SUBSTRACTOR design using logic gate.

RISC :- Any processor having less instructions, less logic gates, less power consumption, less heat dissipation called RISC processor.

CISC :- Any processor have more instruction, more logic gates, more power consumption, more heat dissipation called CISC processor.

* Difference between CISC and RISC :-

CISC	RISC
1. Complex instruction set computing.	1. Reduced instruction set computing
2. Complex and more number of instruction are available Swap inst. → XCH (change) → Complex instruction R_0 R_1 R_0 R_1 (2) (3) (1) temp reg/memory location	2. Simple and less number of instruction are available. MOV destination, source ↑ Simple instruction
3. Agenda is to reduce number of instruction per program 1 instruction → 12 clock pulses (time → 12 μsec)	3. Agenda is to reduce number of clock pulses per instruction. 1 instruction → 1 clock pulse (time - 1 μsec)
4. More number of logic gates is used.	4. Less number of logic gate is use
5. Power consumption and heat dissipation is more.	5. Power consumption and heat dissipation is less.
6. More number of addressing modes are given in CISC.	6. Less number of addressing modes are given in RISC.

7. Die size of processor is larger
Ex:- All intel processors
(8085, 8086, i3, i5, i9 - - -)

7. Die size of processor is smaller
Ex:- ARM, AUR, PIC.

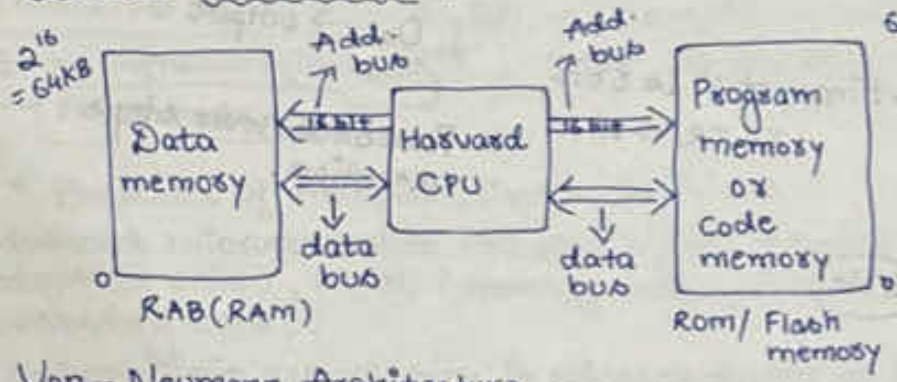
* CISC Instruction → XCHD — change or swap the last nibble

R0 R1
58 45

* Address bus and data bus interface the memory (RAM, Rom, flash memory).

* CPU design policies :-
1) Based upon instruction set (CISC & RISC).
2) Based upon memory organization (Harward and Von-neuman).

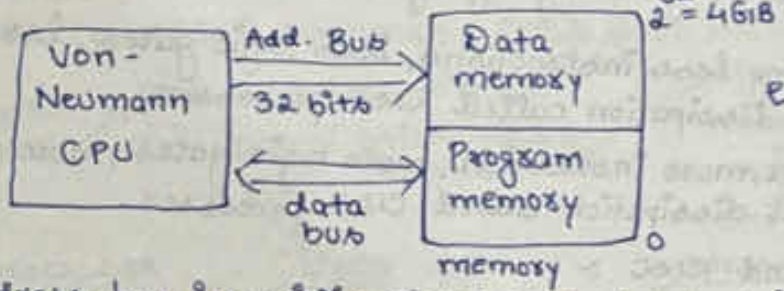
a) Harvard Architecture :-



ex :- AUR - CPU
PIC - CPU
8051 - CPU

mov 30, A
address, data

b) Von - Neumann Architecture :-

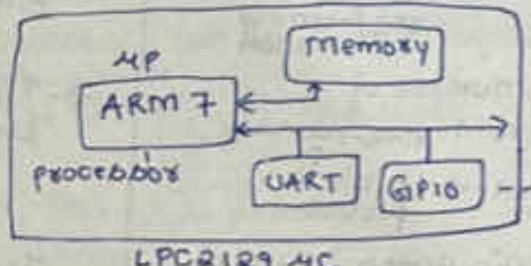


ex :- All intel - processors, memory, ARM 7 - - - -

* Address bus is unidirectional and data bus is bidirectional.
Starting address of memory is 0 upto the 2^{bit} (depends upon address bus)
ex :- 2^{16} ⇒ 64 KB
2^{32} ⇒ 4 GiB.

* Advanced RISC machine (ARM 7) architecture :-
1) Features
2) Block diagram
3) Pipeline architecture
4) Registers
5) Operating modes.

LPC2129 MC - general purpose microcontroller



* Inside the micro-controller, ARM processor is used. That's why they called ARM MC.

Q. What is ARM ?
⇒ The ARM is a processor designed by "ARM Ltd" company.
ARM is not the physical chip, it is the architecture made by ARM ltd.
intellegtual property

* Evolution of ARM processors :-

1st ARM processor is "Acorn - RISC machine" (ARM1),

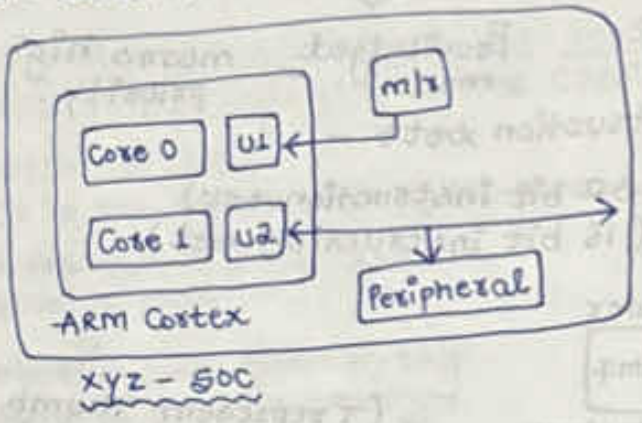
32 bit { ARM1, ARM2, ARM3, ARM4, ARM7, ARM8, ARM9, ARM10, ARM11 → ARM Series

After ARM11 → They started ARM cortex series

1) Series A (application) → Ex :- mobile, smart watch.
2) Series R (Real-time) → Ex :- autopilot
3) Series M (microcontroller)

↑ If fails leads to failed fatal injury

ARM1 to ARM11 → 32 bit processor only.
ARM Cortex Series → 32 bit as well as 64 bit processor.



SOC → System on Chip
↑ customized processor

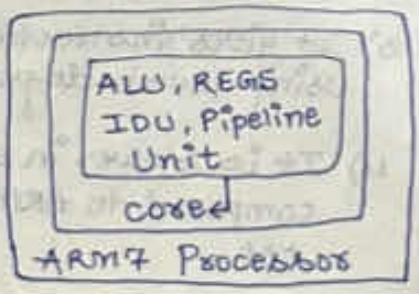
* Core0 perform only 1 instruction at a one time.

* Features of ARM7 Processor :-

1. It is 32 bit - RISC processor designed by ARM ltd company.
2. It is type of Von-neumann.
(↓ In this only one memory is needed).

* Memory split ratio - 3 : 1
3GB Data memory
1GB Program memory

3. Low power and high performance processor
4. Operating voltage is 3.3 volt and maximum supported frequency is 60 MHz
Frequency of crystal oscillator → 12MHz → 12x10^6 osc/sec.
5. It supports 3 stage pipeline architecture
6. Pipeline architecture is used to execution speed of processor.



LPC2129 MC

These are 3 pipeline stages which are -
1) Fetch
2) Decode
3) Execution

7) It support hardware bebug technology, with the help of JTAG we can do hardware bebugging.

JTAG - Joint Test Action Group.

8) It supports load and store instructions for faster memory operations.

9) It has single clock cycle instructions. (clock pulse)

10) Same size instructions.

11) It has total 37 number of registers.

12) ARM 7 works into 7 different modes.

7 mode - User, Supervisor, FIQ, IRQ, undefined, Abort, System

User — Non-priviledged mode

System, Supervisor, FIQ, IRQ, undefined, Abort — Previledged mode — means high priority.

13) ARM7 supports different instruction sets -

- ARM Instruction Set (32 bit instruction set) ← default
- Thumb instruction Set (16 bit instruction set).

Program (using thumb Instruction) → compile → hex.file

older: ARM7 — Architecture name: ARMV3

Newer: ARM7 — Architecture name: ARMV4T (T represent Thumb instruction set)

{ Command :- uname -m → (Print architecture name)

* Advantages of Thumb over ARM instruction set :-

ARM instruction Code

1) 1 — instruction 1
2 — " 2
3 — " 3
4 — " 4

Code size :- 4 x 4 = 16 bytes (4 instruction)

2) At run-time arm instruction code directly go to the core.

Faster in execution

3) It gives direction instruction to the 32 bit core.

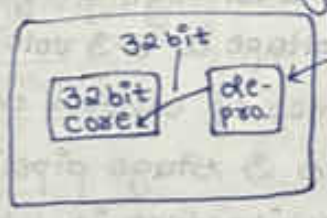
4) It is faster in execution.

Thumb instruction code

1) 1 — instruction 1
2 — " 2
3 — " 3
4 — " 4

Code size :- 4 x 2 = 8 bytes (4 instruction)

2) At run-time, thumb instruc. go to the thumb compressor and then converted 16 bit to 32 bit and then go to the core.



3) It gives instruction through as using thumb de-processor.

4) It is slower in execution as compared to ARM instruction set.

* ARM 7 modes :- Arm 7 is a processors and works in 7 different modes.
Modes are selected automatically by processor.
We are not suppose to change the modes.

Processor will change the mode in the following three reasons :-
1) Interrupt (FIQ / IRQ interrupt).
2) Exception (Predefined error condition)
3) Reset.

Advantage of using or giving the mode is data security.
When processor is interrupted, then automatically it call the ISR function.
ISR - Interrupt Service Routine.

* Every processor mode except user mode can change the mode by writing directly to the mode bits of the CPSR.

1) Supervisor mode :-
1. It is the default mode of the processor after reset.
* Supervisor mode 1st execute startup code
↓
startup code given by chip designer can be customized

Startup code do processor or microcontroller initialization

① Startup code/ process of embeded software.
After reset
↓ 1st execute
Startup code
↓ 2nd execution
main function

local variable → declare in stack section
global variable → declare in data section
↓
global variable initialize by '0'.

② Startup process of ARM processor (H/W)
After reset
↓ ①
Supervisor mode → ② startup code
↓
user mode ← ③ = ④ → main function

= ↓ last line
= ↓
= ↓
last line move to the user mode.

2) User mode :-
1. It is non-priviledge mode
2. It is normal program execution mode
3. This mode is selected automatically.
4. Most of the time processor is in user mode.

```
main()
{
  =
  =
}
hello()
{
  =
  =
}
```
Normal Program

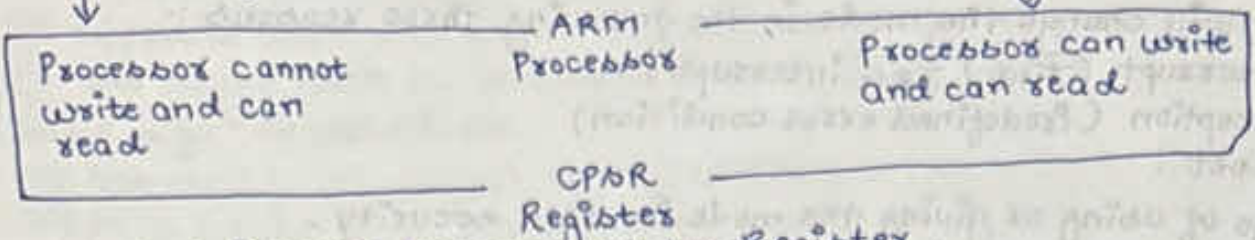
```
ISR()
{
  =
  =
}
```
Special Program/ high priority program.

(only allows read access to the control field in CPSR)

* Non-priviledge mode

(allows read write access to the CPSR)

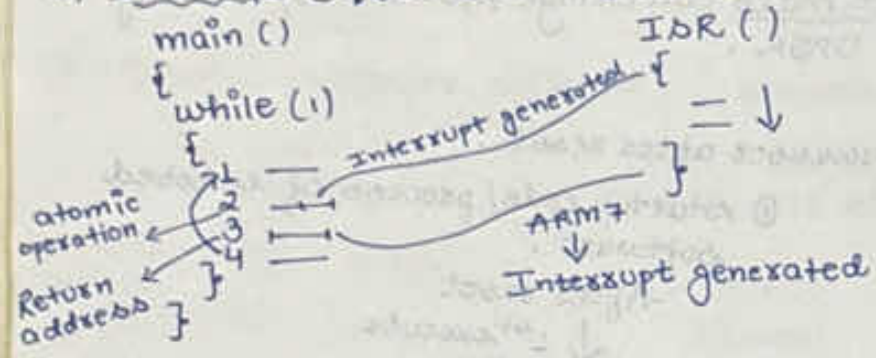
Priviledged mode



CPSR → Current Program Status Register.

3) Faster Interrupt Request (FIQ) mode :-

4) Interrupt Request mode (IRQ) mode :-



Atomic operation :- instruction in execution called as atomic operation or instruction or execution which we cannot break in middle.

IDR store the return address. Hence after executing the IDR, processor will go to next instruction.

* Every action is stored in memory.
* If line number 4, the return address will go to while (1).
* Return address store in the stack and stack is the part of RAM.
* Interrupt must have priority. Processor cannot execute two function/ two instruction at a time.
* FIQ has high Priority Interrupt than IRQ mode.

(less priority)
IRQ mode

User mode

(Higher priority)
FIQ mode

* Before changing the mode, processor will store the current code to another (anywhere) or in SPSR

SPSR → Saved Program Status Register

↓

This register is used to store the information / current of previous mode

* In IRQ mode, if FIQ mode interrupt generate then it will go to FIQ because FIQ has higher priority and

If FIQ mode, if IRQ interrupt generate it will first execute FIQ mode and then execute IRQ mode because IRQ has less priority.

5) Undefined mode :- This mode is selected when any undefined instruction is fetched by the processor.

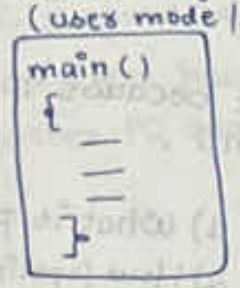
In older version of ARM7, Thumb instruction code is undefined, then the processor will go to undefined mode and print the error msg.

6) Abort mode :- This mode is selected if the processor fetches the data / instruction from the reserved memory location.

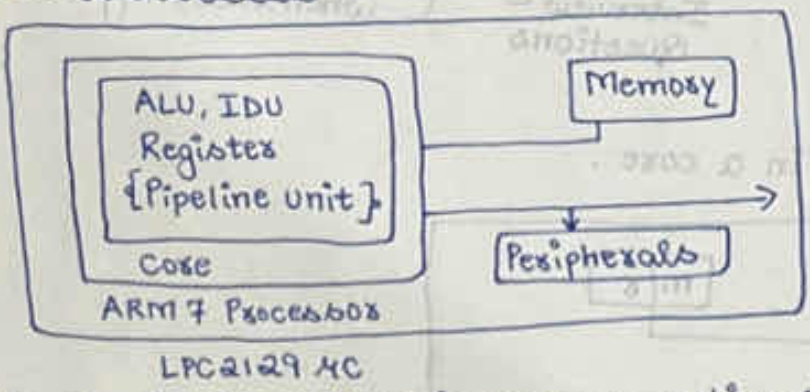
7) System mode :-

- This is priviledge mode of user.
- Normal programs can be executed in the system mode.
- Writing wrong value in CPSR, it will get affected when it is in the non-priviledge mode or it may hault the execution or give the error.

(Non Priviledged / Previledged)
(user mode / system mode)

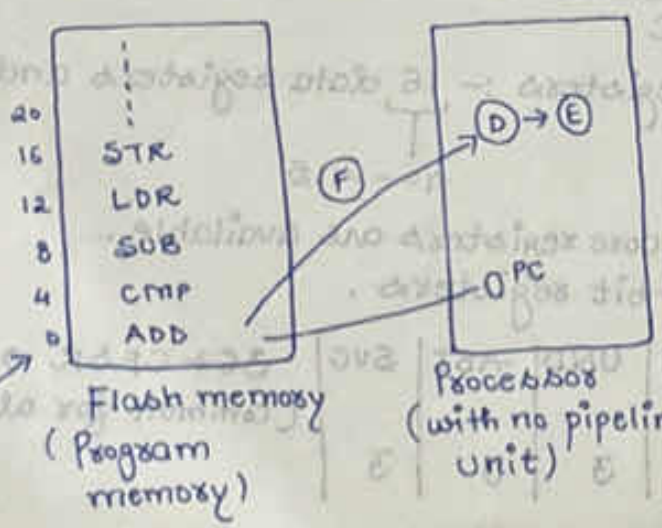


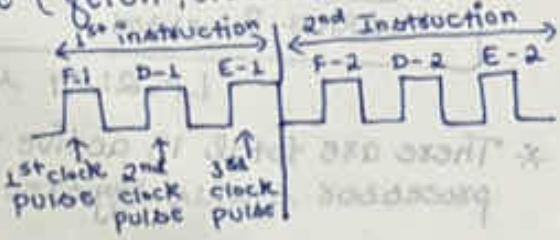
Pipeline Architecture :-



LPC2129 MC

- This pipeline is used to increase execution speed of processor by executing an instruction while other instructions are being fetched and decoded.
- Processor do step by step instruction.
- Pipeline architecture is 3-stage pipeline (fetch, decode and execution).





F - 1 → Fetch 1st instruction
D - 1 → Decode 1st instruction
E - 1 → Execute 1st instruction

— PC (Program Counter) is always pointing to program memory.

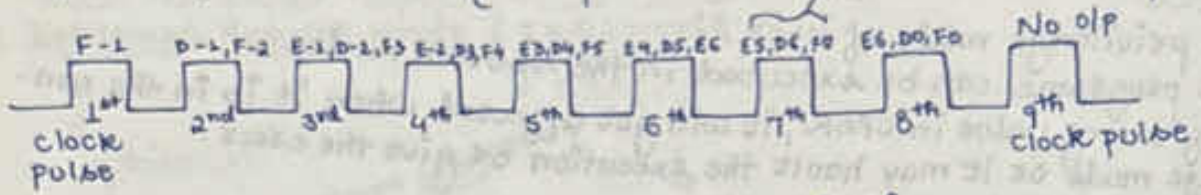
For 1st instruction → 3 clock pulse is used.

1) With no pipeline unit :-

* 18-clock pulse for execution in with no pipeline unit.

2) With Pipeline Unit :-

— Parallel work done by the processor. → instruction level parallelism



* 8-clock pulse for execution in with-pipeline.

— Here processor do parallel work.

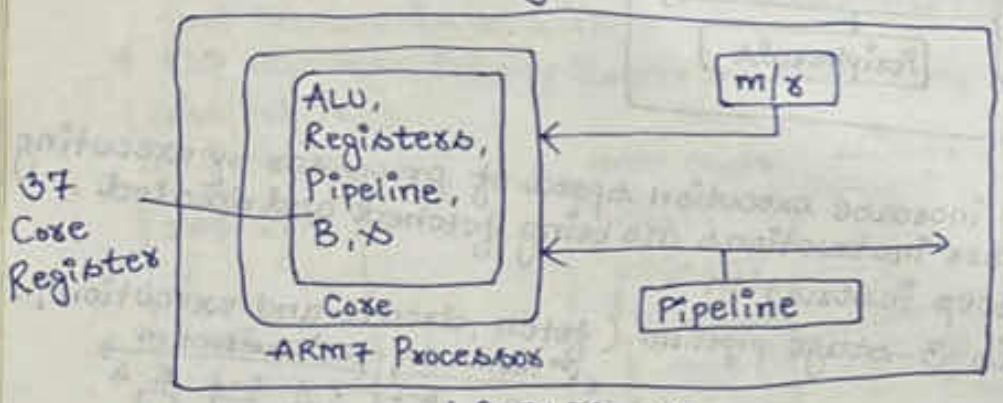
— Processor will use less nomber of clock pulse because to execute more instruction when pipeline unit is present.

ARM9 :- 5-stage pipeline is used.
ARM10 :- 6-stage pipeline is used.

Interview Questions
1) What is pipeline ?
2) How pipeline is worked ?
3) How interrupt is generate when executing instruction?

* ARM7 Register set :-

* Total 37 Registers are given in a core.



LPC2129 μC

* There are total 18 active registers :- 16 data registers and 2 processor status register.
r0 - r15

① Total 37 numbers of core registers are available.
② All registers are 32-bit registers.

mode → uses/system	FIQ	IRQ	UND	ABT	SUC	36 + CPSR → 37 Registers. (Common for all modes)
No. of Register → 16 Registers	8-Register	3	3	3	3	

* List of user mode Registers :-

R0
R1
:
:
R12
} General Purpose register (used to store data / address)

R13 – (Stack Pointer)
R14 – (Link Register)
R15 – (Program counter)
} assigned to a particular task / special function.

Types of Register in any microcontroller { General purpose register – given inside processor. (GPR & SFR)
Special function register – given inside RAM / memory.

* Special function register is used to hardware communication and general purpose register is used to store data and address.
* Depending upon the task, R13 and R14 are also be used as general-purpose registers.
* Every register has address.
  Whatever PC store that take as address.