

Embedded Systems

① ARM micro processors → ACORN RISC Machine.



- (i) features & basics. [Reduced Instruction set computer]
- (ii) architecture.
- (iii) programming model.
- (iv) instruction set.
- (v) Thumb mode.

ARM → RISC based load & store architecture, fast.

32 bit processor.

- kind high end applications.

- ARM supports high code density - complex application.
- low power consumption.

features → load & store architecture → 32 bit processor.

supports 2 modes thumb mode Jazelle.

↓
16 bit

↓
8 bit

→ Barrel shifter → parallel processing can be done.



which can maximize hardware usage available on chip.

→ conditional execution - instruction gets executed only when condn is being met, which maximizes execution throughput.

- 1) 16 registers - (R₀-R₁₅) | destination Register - R_d.
- 2) ALU gets input from Registers R_m & R_n.
- 3) MAC - Multiplication : accumulation.
- 4) Instruction decoder - helps in decoding instruction register.
- 5) Incremental will be every time increment to next location.
- 6)

* Three kinds of instruction set supported in ARM core.

(i) ARM (ii) Jazelle (iii) Thumb .

* CPSR - with CPSR one can set operation state & accordingly instruction set can be selected .

* R_m & R_n values will be fetched from buses A & B & computation carried out in ALU . address registers are used to hold address & address bus will facilitate the storage action .

ARM - modes

② CPSR - Current program status Register, it will help in changing modes.

(i) privileged mode - wide level of access.

(ii) non privileged mode - not as privileged.

① abort mode - when there is memory access failure.

② fast interrupt request mode \rightarrow both when interrupt occurs.

③ interrupt request \downarrow privileged mode.

④ supervisor mode \leftarrow Reset or Restart.

⑤ system mode - similar to user mode & diff is it have complete access to

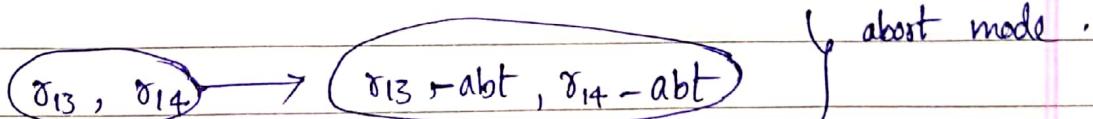
⑥ undefined mode \leftarrow when process encounters undefined instr.

⑦ user mode \rightarrow non privileged mode (because of CPSR). without restriction.

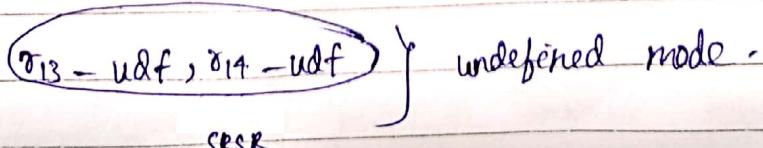
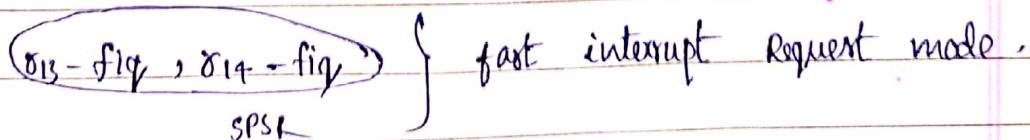
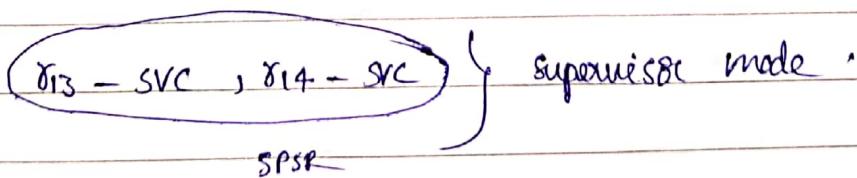
used by programmer for programs & applications.

CPSR :

\Rightarrow Register banks - (Mode switch).



SPSR - saved program status Register.



SPSR : Saved program status Register

- specially meant for privileged modes of operation.
- System mode does not support SPSR.
- SPSR is used to hold status of CPSR. so, it is called storage Register which can hold current state of CPSR.

ARM Register Architecture :-

→ Load & store → Register plays basic role.

16 register but

13 general purpose Register & 3 special function Register

(r_0 to r_{12}),



r_{13}, r_{14}, r_{15}

- used for any operation as wish of user.

i) stack pointer

ii) link Register

iii) program counter

(i) $r_{13} \rightarrow$ stack pointer, it helps you to hold the address of top of stack.

(ii) $r_{15} \rightarrow$ program counter, holds address of next instruction to be executed.

(iii) $r_{14} \rightarrow$ link register, used for fetching address back from where control of execution has been initiated.

CPSR :- Current program status Register

when result of ALU is -ve

N → negative flag

N Z C V	Unused	I F T	Mode
---------	--------	-------	------

Z → zero flag → set when result of ALU is 0000.

C → carry flag → set when result of ALU is carry generated.

V → overflow flag → set when result of signed operation is an overflow.

IF → interrupt mask.

T → Thumb mode.

Mode → one of 7 modes can be selected.

with I → one can mask or unmask IRQ interrupt.

with F → one can mask or unmask FIQ interrupt.

ARM states:

- ① default \rightarrow ARM - 32 bit { 16 bit & 8 bit}
- ② Thumb \rightarrow 16 bit
- ③ Jazelle \rightarrow Java enabled operation

J = 0 } not enabled

T = 0 } Thumb not enabled

Pipe line

fetch { 3 stage pipe line
decode
execute

\rightarrow for faster processing

5 stage pipeline

fetch - it will fetch instr

decode - decode instr

execute - execute instr

buffer - ALU results will be buffered for clock cycle

write - get results

Memory Management - ARM (cache usage)

- * larger the memory, slower it gets.
- * cache → small, fast component

Temporal locality

Spatial locality

cache hit - if you find content

cache miss - if you can't find content

③ AMBA - Advanced Micro-controller Bus Architecture:

- meant for bus. (taking data)

- used in micro controller bus.

- open standard, on chip inter connect specification for connection management of functional blocks in the system on chip.

- AMBA helps in getting design in 1st attempt with right design.

- enables usage of large no of peripherals as well.

⇒ it is used, when there are multiple processor, it is real complex.

AMBA standards:

* AHB - advanced high-performance Bus - used for high band-width

* ASB - advanced system Bus } interface b/w elements

* APB - advanced peripheral Bus - used for low-band-width interface.

ATB - advanced Trace Bus

AXI - AMBA extensible interface

* Bridge → AHB, ASB and APB are separated with bridge.

① APB: (low band-width peripherals) - used for connecting.

- used for read/write from the bridge to peripherals.

- Burst transfer not possible.

② AHB: used for connecting higher band-width peripherals.

- Multi master ie multi slaves are supported.

- Burst transfer supported.

③ AHB-lite: - advanced version AHB.

- single master

- no complexities

④ AXI - used when high band-width, low latency is requirement

- update of AHB

- supports burst mode transfers

- supports pipe lined transfers

⑤ AXI-lite:

simplified version of AXI

- no support for burst data transfers

⇒ Bus:

- Bus is mechanism by which CPU communicates with memory and IO devices.

- Bus defines protocol for communication.

1) asynchronous communication protocol :-

address bus → ↗ ↘

eg: UART - Universal Asynchronous Receiver cum Transmitter.

data bus - ↗ ↘

2) synchronous communication protocol :-

control bus - ↗ ↘

eg: I2C

status bus - ↗ ↘

⇒ Bus characteristics :- 1) Address & data lines may be multiplexed.

- every device on bus must be able to drive max bus load.

- Bus may include a clock signal.

ALP :-

- (1) entry directive - tells location of first executable instruction.
- (2) Area - defines storage area.
- (3) end directive - to show the code is getting completed here.

LPC 2148

- 1) AREA RESET, CODE, READ ONLY . | AREA PROGRAM, CODE, READ ONLY .
- 2) ENTRY
- 3) LOOP .
- 4) END .

MRS - special command .

- used to Read Register content of CPSR .

ARM Memory Organization :-

loops - Temporal locality .

sequence of code - spatial locality .

* cache can store data & address .

* amount of time that it took to really go to next level in cache penalty (Miss penalty).

How to map the cache to primary memory?

(1) direct mapped

(2) set associative

(3) fully associative

①

data
tag

0	1	2	3	4	5	6	7
12							

search

$$12 \bmod 8 = 4$$

②

data
tag

0	1	2	3
12			

↑

$$12 \bmod 4 = 0$$

② could be in any of 8 blocks

data

0	1	2	3	4	5	6	7
12							

tag

9

9

↑

↑

↑

↑

12

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

Assembler directives in ARM E

- help in assigning memory for programs
- Referencing
- some quick housekeeping exercises.

① constant

② equate

③ Area

④ storage

constant :

allows programmer to enter fixed data into program memory

- it can be name, message, command, look up table.

- The data treated as permanent part of program.

DCB (define constant Byte) - 8 bit numbers

DCW (constant word) - 32 bit numbers 8192 addresses

DCD (constant data) - can be used for DCW

TTL → title

ALIGN

EQU - equate

Memory Organization

① Temporal Locality:

If particular data location is referenced, then there is a chance that it will be referred again soon.

② Spatial Locality:

If particular data location is referenced, then locations nearby shall be referred in near future.

Eg: array.

Main memory \rightarrow RAM (implemented as DRAM).

Cache \rightarrow nearer to processor (implemented as SRAM).

Largest to smallest is hard drive.

Ground Rule:

- faster memory is closer to processor & most expensive.
- slower memory is far to processor & less expensive.

\Rightarrow The unit of information present at not in block.

~~hit rate / hit ratio:~~

- It is fraction of memory access found in upper level.
- It defines performance.

miss rate:

- It is fraction of memory access not found in upper level.

- It can be used for performance measurements.

hit time:

- time to access the upper level of memory hierarchy & the time needed to determine whether access is a hit or a miss.

upper level = faster = smaller hit time 😊

Cache: a safe place to store things.

① direct mapped:

each memory location is mapped directly to one location in cache.

- This is very simple.

(Block address) modulo (no of cache blocks).



byte address

bytes per block .

(block number) modulo (no of sets in cache)

for match.

all blocks in set are searched

- ② set associative → a block is directly mapped into set, so then
③ fully associative → all entries in cache must be searched.
because a block can be placed in any one.

NGIX

PINSEL Register

- (1) PINSEL0 → P0.0 to P0.15
(2) PINSEL1 → P0.16 to P0.31
(3) PINSEL2 → P1.16 to P1.31

P0 → 30 pins

P1 → 16 pins