M. Vaidehi
Asst. Prof
ISE- Dept

ARM Processor → FEATURES.

LPC 2148 – 64 pin IC.

The ARM is a 32 bit processor

(1) It has a 32 bit ALU

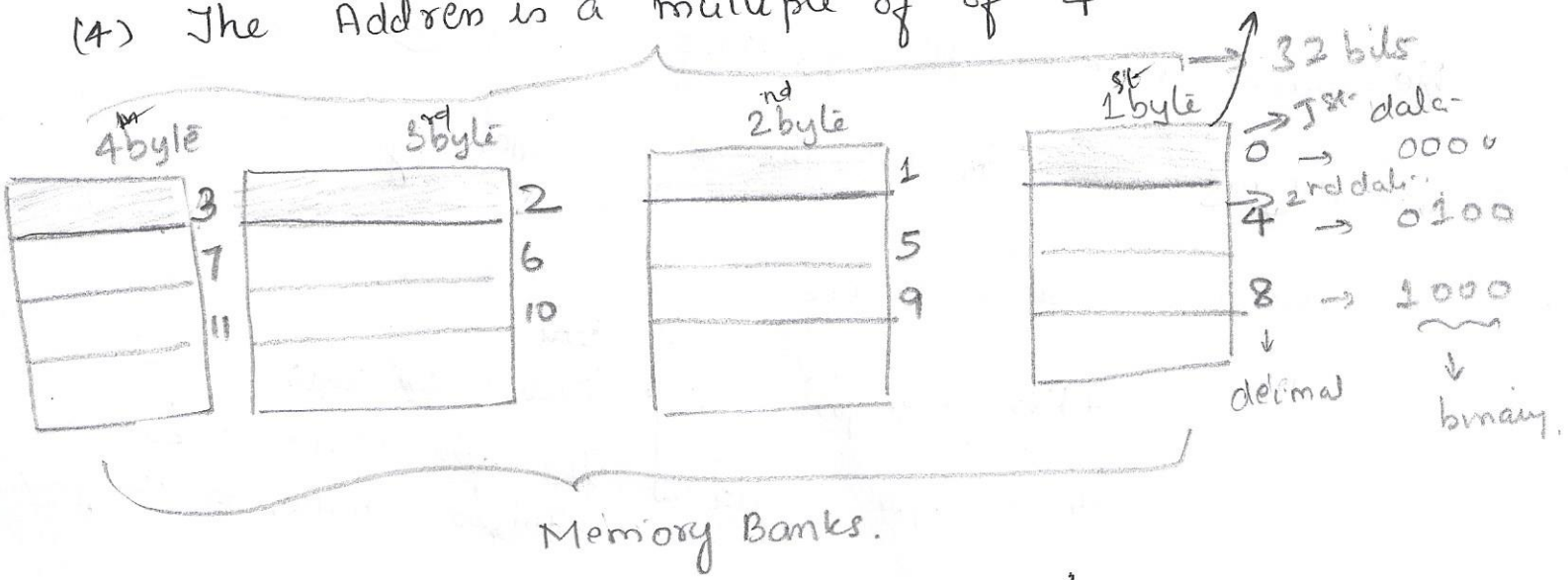
(2) 32 bit Data bus

(3) 32 bit address bus

It can transfer 32 bit data in one cycle

one cycle / clock cycle.

(4) The Address is a multiple of of 4



Each instruction/data is 32 bit in size, /e

4b The data is stored in aligned form

So that the operation can be performed in one cycle.

(7). Legacy

(5) It is 32 bit instruction.

* All instructions of same size
* Every instruction is fetched in one cycle.

(6) All instructions are stored in aligned form.

(7) It has 32 bit addres bus. So it can address 2^{32} → memory location ie: 4GB.

(8) It ~~is~~ is designed based on the Von Neumann Architecture → here instruction and programgram are stored in the same memory in Harvard architecture

64 KB for program
64 KB for Data

} Based on this the 8051 – 8 bit micro controller is designed

ARM	8051
(1) 32 bit controller	(1) 8 bit controller.
(2) Von Neumann Architecture	(2) Harvard architecture.
(3) Data format (bits) 8-bute	(3) Data format.. 8 bits → Word

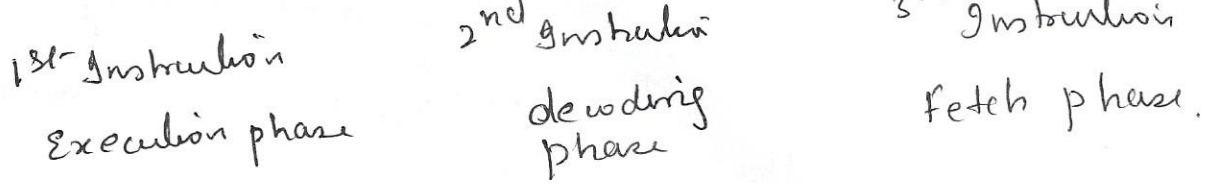
Note: Word refers to the size of data handling capability of the processor (in 8086 word - is 16 bits)

(9) It has 3 stage piplene → this makes the processor to operate at a higher speed

Example: 1st instruction is fetched and decoded,

During the decoding period of 1st instruction the 2nd instruction is fetched

Now when the 1st instruction enters the execution phase then the 2nd instruction enters the decode phase and at the same time the 3rd instruction is fetched.



(10) It follows the load store model.

Here no data manipulation is done in memory all data manipulation is carried only regis using registers.

like in 8086 you have

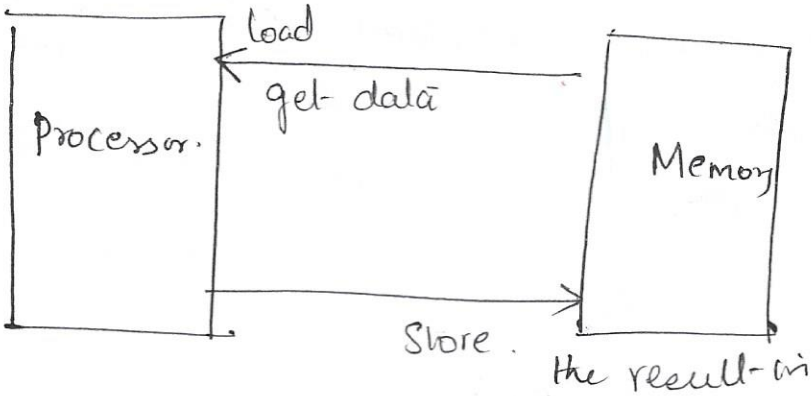
ADD AX, [1245H]
↳ memory

one of the operand is memory in ARM this is not possible all operands use only the register.

Example add r1, #08

mov r2, r1
↳ Data transfer.

But only the ~~Loc~~ load store instruction are allowed to access the memory



No data ~~Da~~ manipulation is done directly. in the memory.

(10) Zoo Screen shar.

(11) The ARM procesor will be in either of the 7 mode.

(1) Supervisor mode ← when a user switches on

(2) User mode → while programing el.

(3) System Call mode

(4) FIQ mode → fast interrupt mode

(5) IRQ mode → Normal Interrupt IRQ mode.
→ Software interrupts.

(6) Undefined mode

(7) Abort mode.

(12) ARM (Advanced RISC Machine) has Seven Addressing modes.

(1) Immediate ex: mov r1, #02;
add r1, #04;

(2) Register Ex: mov r1, r2;

(3) Direct

(4) Register Indirect

(5) Register relative

(6) Base Index

(7) Base with scale index

Features of ARM Procesor Architecture:

1. It has 37 registes all of 32 bit in the register file
   - 20 → are hidden / banked and are useedin privileged mode.
   - 17 → are used in user / system mode
     ie r0 - r15, CPSR [Current Program Status Register]
2. There are 7 procesor mode (six privileged, abort, fast interrupt, request, interrupt request, Supervisor, System and undefined, one non privileged mode user.
3. It has load store architechture.
4. 3 operands instruction.
   Two source operand registers and one result register.
5. Conditional execution of instruction
6. A very powerful load and store multiple register instruction.

7. It can perform any ALU operation or a shift operation in a single clock cycle.

8. Open instruction set extensions through the co-processor instruction set including adding new registers and data types to programmers model.

9. A very dense 16 bit compressed representation of the instruction set in the thumb architecture.

10. It supports a tool ~~kit~~ ~~is~~ kit, which includes instruction set emulator for hardware modelling and software testing and bench masking as an assembler.

11. There are six versions of ARM (V1 to V6)
V4 includes Thumb instructions.

12. Enhanced multiply instruction ie 16x16.
it has a (MAC) → faster multiply Accumulates.

Registers of ARM Processor: (Programmers Model)

(1) register set r0 - r15

(2) CPSR [Current Program Status Register (CCPSR)]

(3) SPSR [Saved Program Status Register]

→ It has 32 - bit capacity register, it holds data or an address

The registers are identified using the letter 'r' prefixed to the register number like r0, r1, r2 .. r15

These registers r0, r1, ... r15 can be used in user mode operations.

→ r0 to r15 are visible to programmers these are data registers

→ CPSR and SPSR are also visible.

→ r13, r14, r15 are assigned a particular task.

r13 → SP (stack pointer)

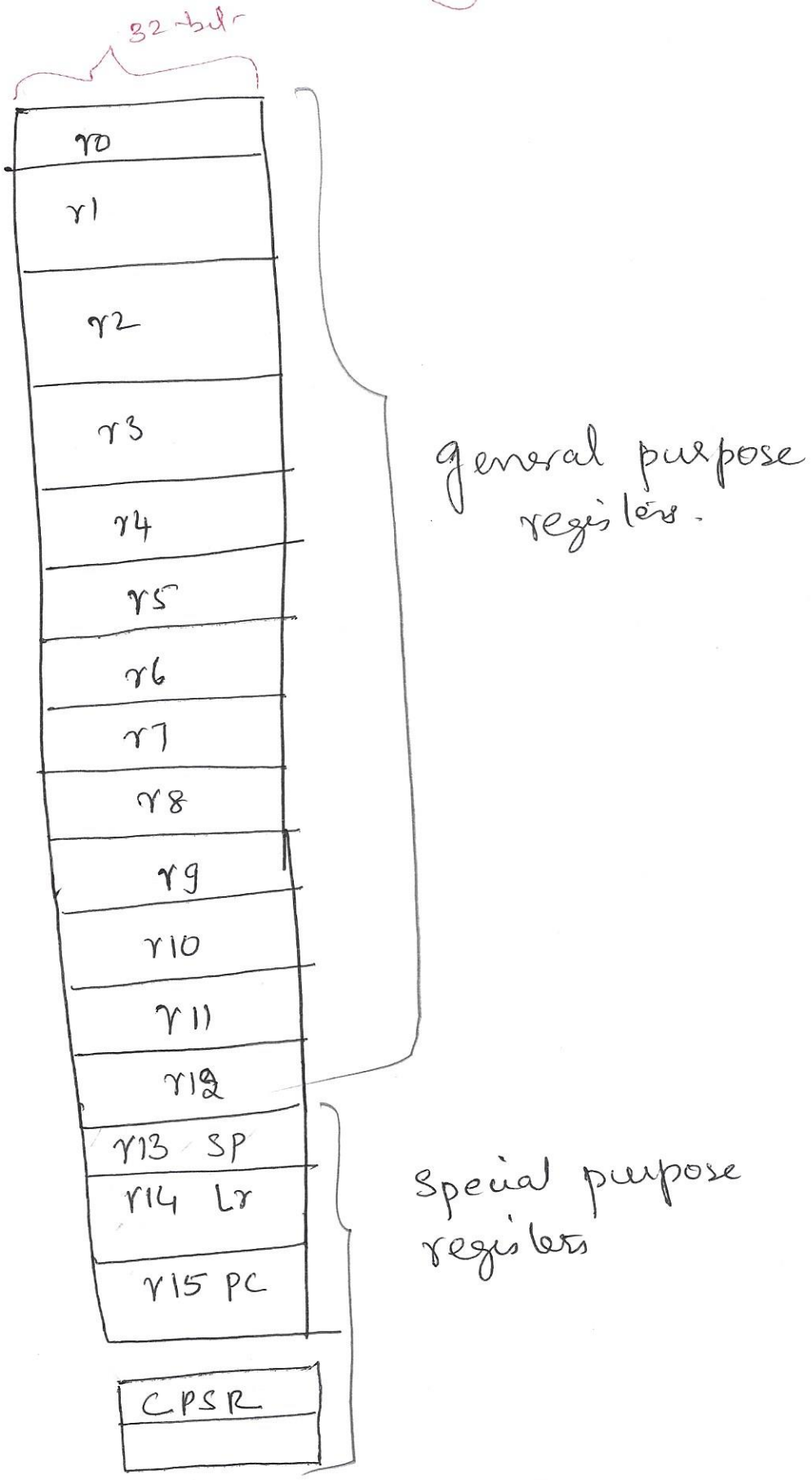
(lr) r14 → lr (~~interrupt registers~~ Link register)

r15 → P.C. (Program Counter)

r13: used as stack pointer (r13 sp) to store Stack of stack in the current processor mode and also can be used as a general purpose register except when processor is running in operating system mode.

r14: used as a link register (r14 lr) → it stores the return address when ever it calls a subroutine and can also be used as a general purpose register.

r15: used as a program counter (r15 pc) to store address of next instruction to be fetched by the processor and it is also used as a general purpose register



ARM Register Set