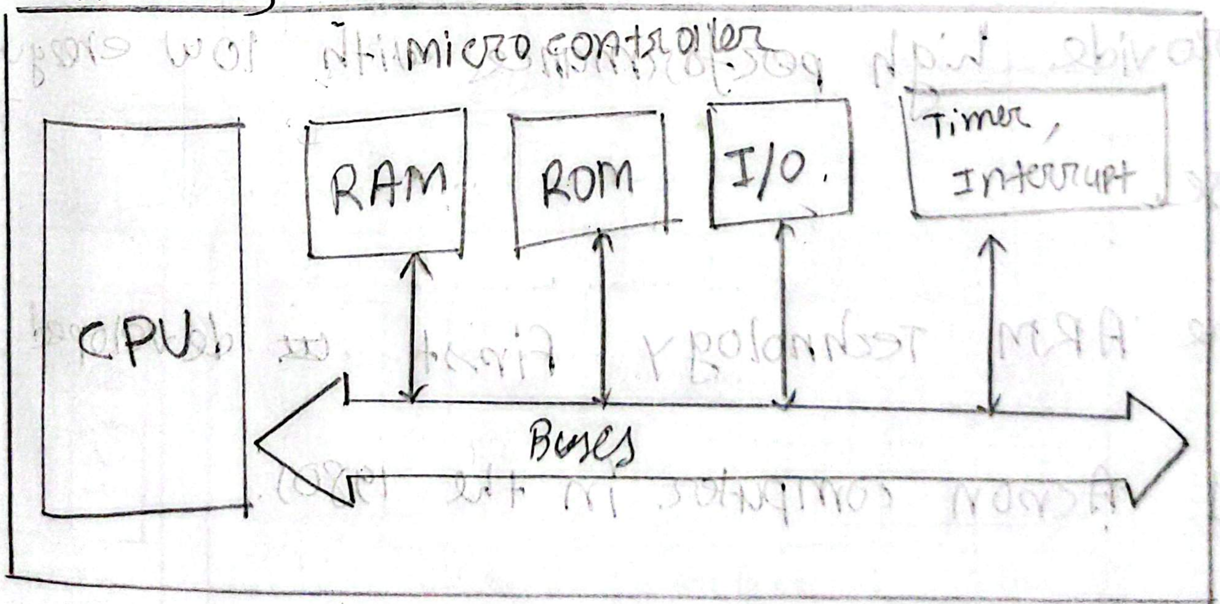
Microcontrollers

A microcontroller is a computer. Integrated circuit that contains a CPU, RAM, Rom and I/O peripherals on a single chip

It is often called a MCU (microcontroller unit) and it is designed to perform specific control task.

Since microcontrollers are cheap and small, they are widely used in many devices.

2.9,



Simplified view of the Microcontroller.

ARM

ARM stands for Advanced RISC machine

It is a type of microprocessor architecture based on the RISC (Reduced Instruction set computer) design, which makes it ~~first~~ fast, efficient, and low-power.

ARM processors are widely used in mobile tablet and embedded systems because they provide high performance with low energy use.

The ARM Technology first ~~as~~ developed by Acorn computer in the 1980s.

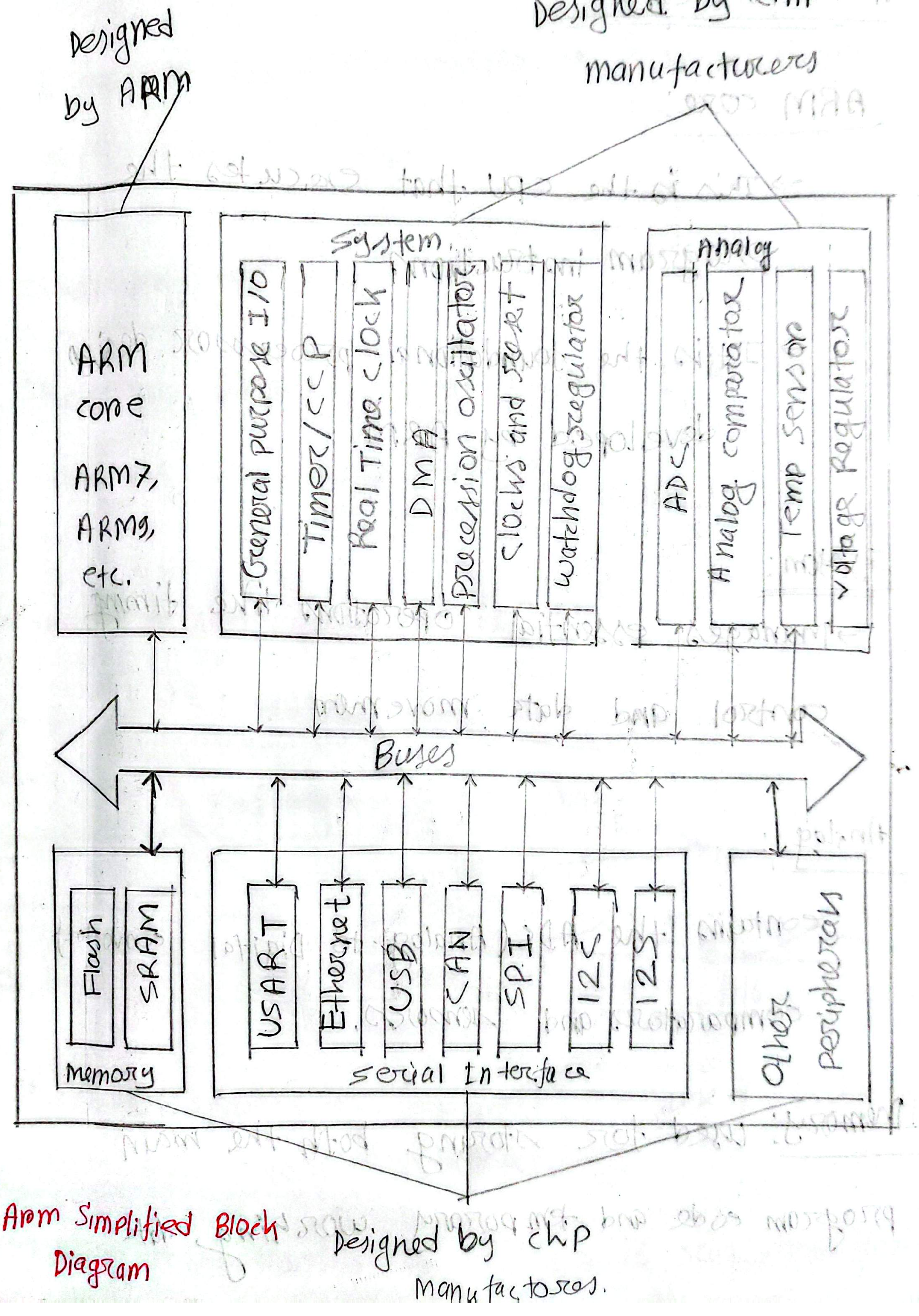


Fig: Arm Simplified Block Diagram

Architecture explain

ARM core:

→ This is the CPU that executes the program instructions

→ It is the foundational processor design developed by ARM

System:

→ Manages essential operations like timing control and data movement

Analog :-

contains the ADC (Analog to Digital convertor) comparators and sensors.

Memory: used for storing both the main program code and temporary working data

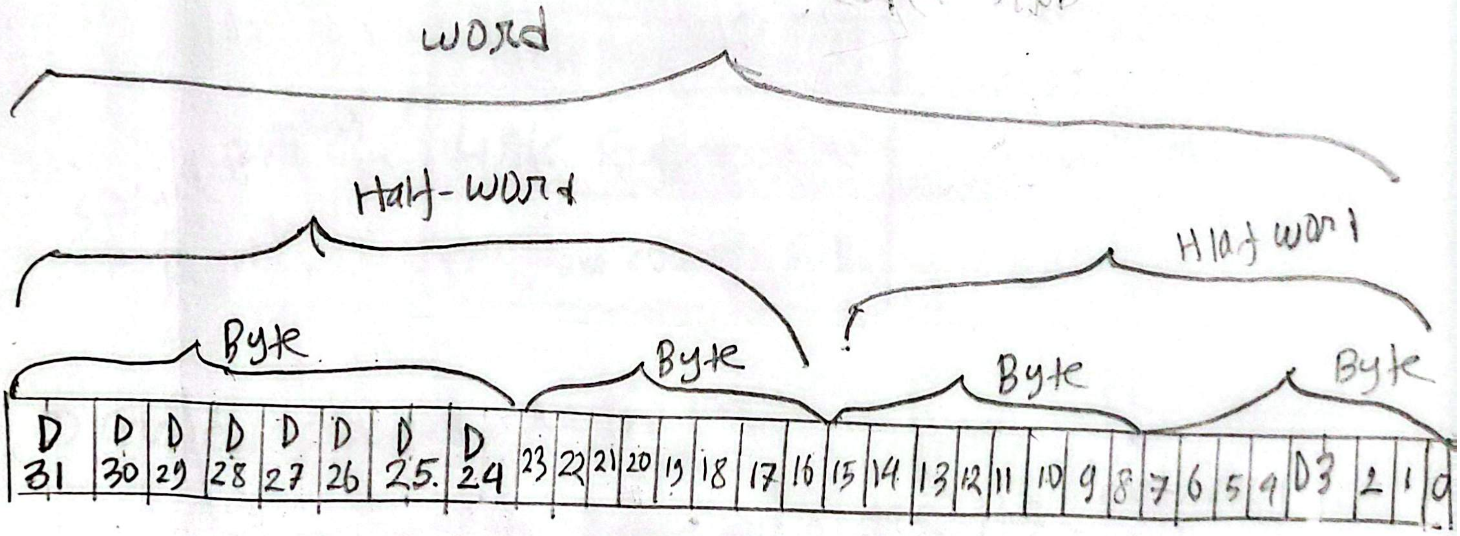
Serial interfaces

Allows to the microcontroller to communicate with external devices.

Other peripherals:

These are usually designed for I/O functions and to add unique features.

General purpose Registers in the ARM



→ In an ARM CPU registers are used to store data or addresses temporarily.

→ ARM microcontrollers contain 16 registers for arithmatic and logic operation

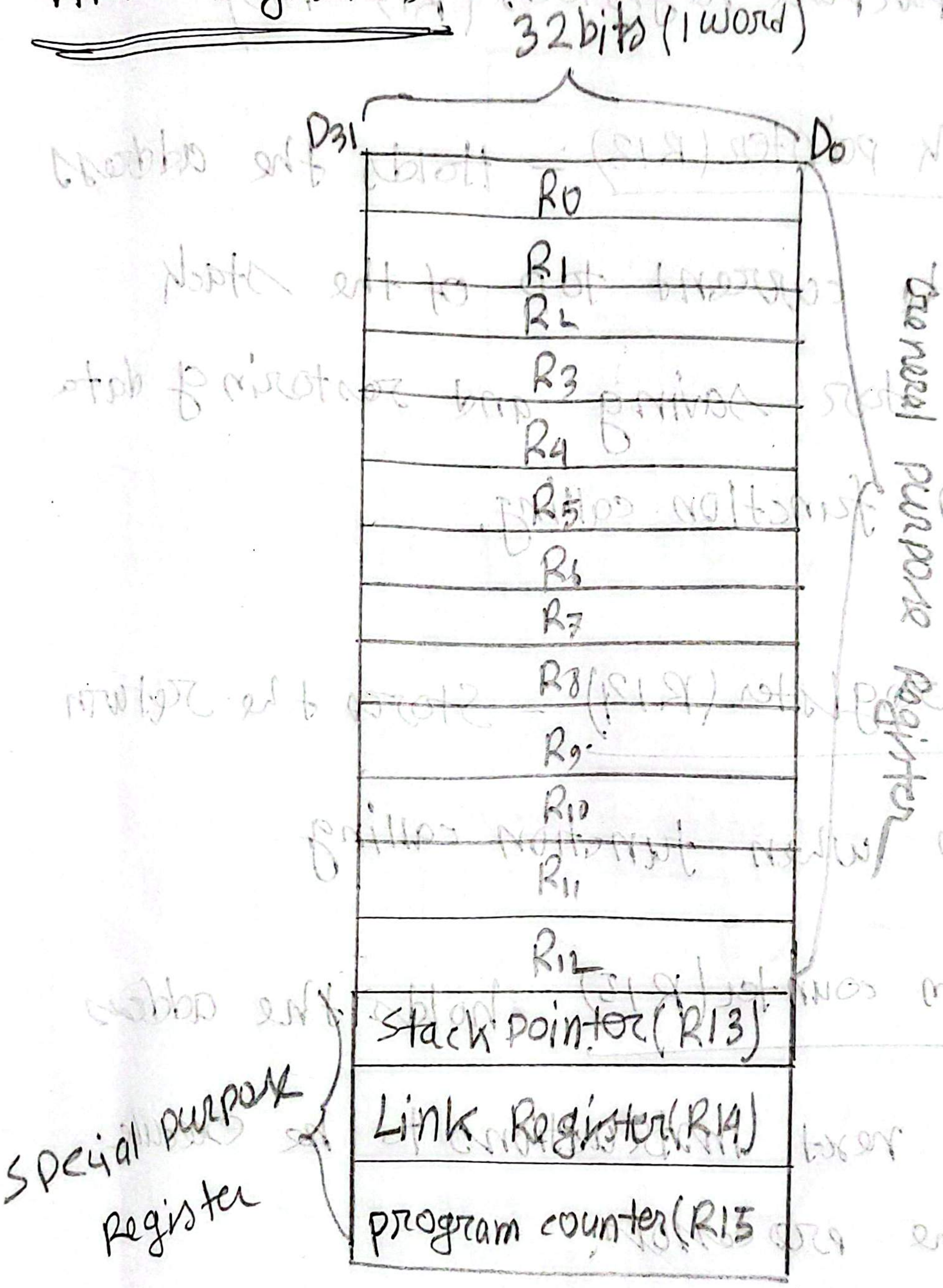
→ Each register is 32 bits wide (D31 to D0)

→ the Default data size in ARM is 32 is called a word.

→ here half-word means 16 bits and byte means 8 bits.

* Arm supports byte, half-word, word data types.

ARM Registers



① General purpose register :-

- These 13 registers are the primary workspace for the CPU

- They are used to hold data variable and intermediate calculation results.

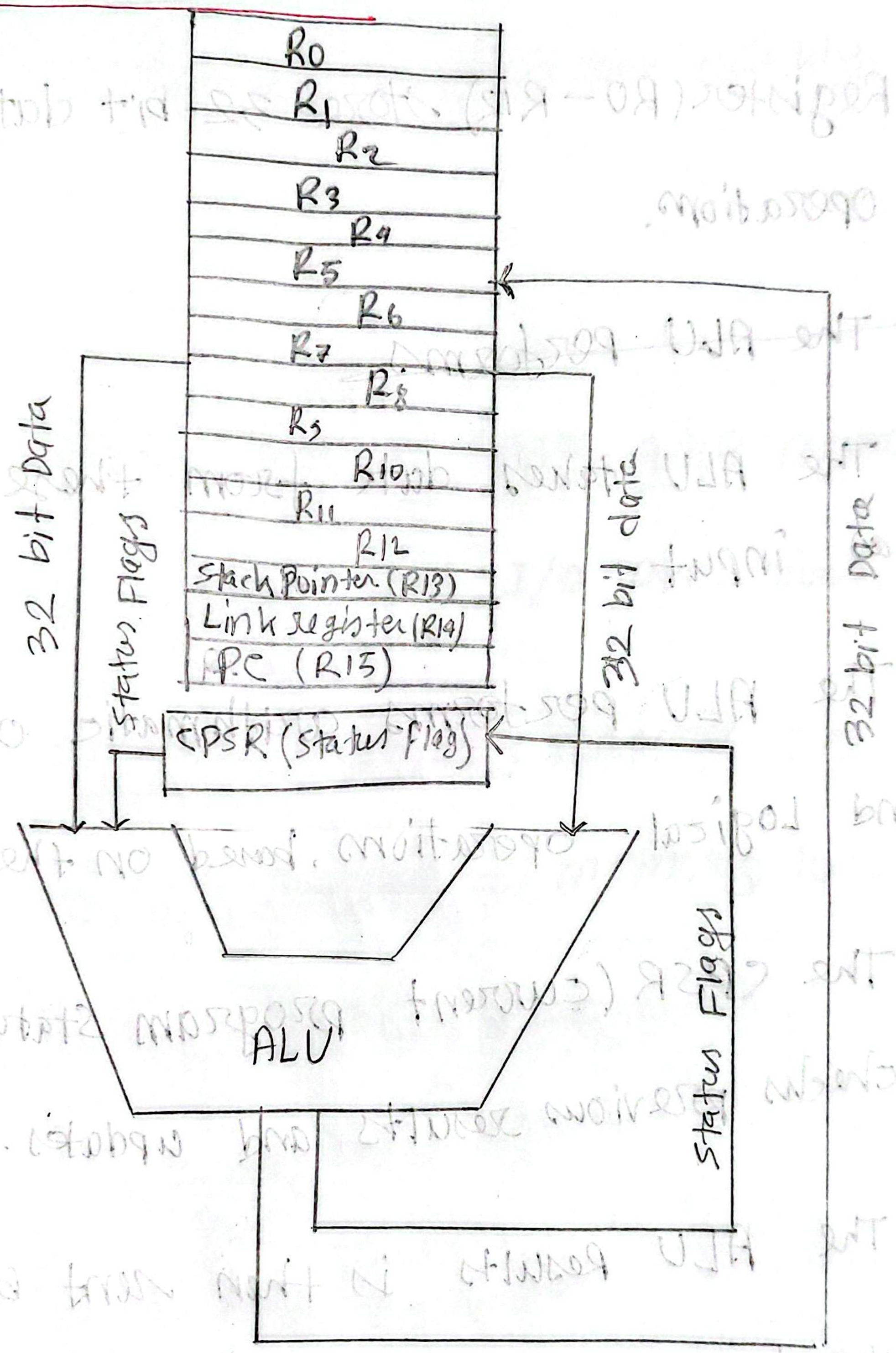
② Special purpose Registers (R13 – R15)

Stack pointer (R13) :- Holds the address of the current top of the stack used for saving and restoring data when function calling.

Link register (R14) :- Stores the return address when function calling

Program counter (R15) : holds the address of the next instructions to be execute by the processors.

ARM Registers and ALU



Working procedure of ARM Registers and ALU

(i) Register (R0 - R12) store 32-bit data for operation.

(ii) ~~The ALU performs~~

(ii) The ALU takes data from these Registers as input

(iii) The ALU performs arithmatic operations and Logical operations, based on the instruction

(iv) The CPSR (current program Status Register) checks previous results and updates.

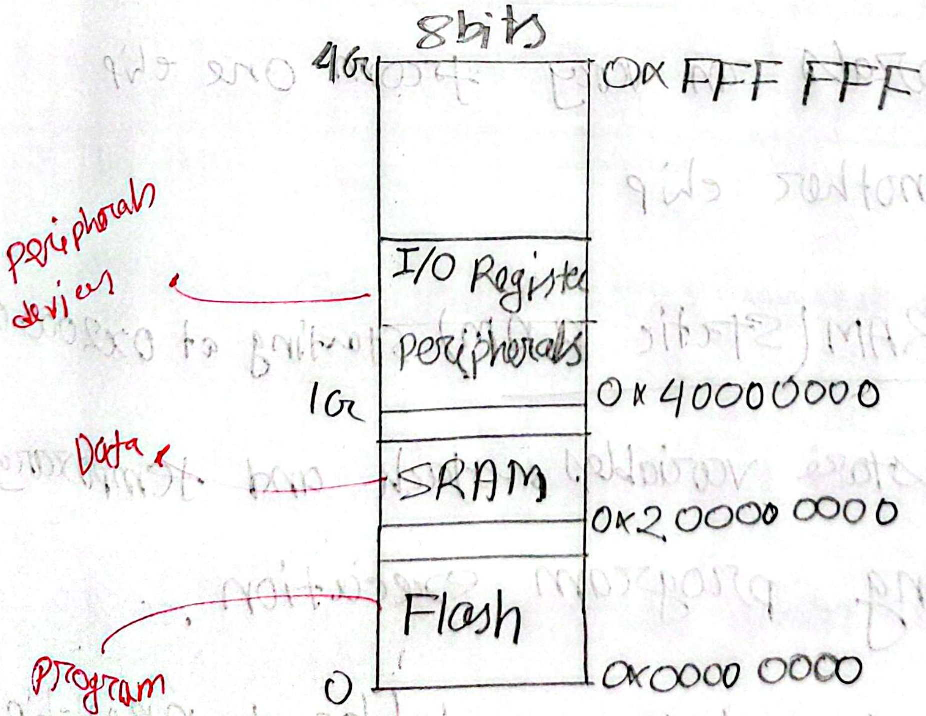
(v) The ALU Results is then sent back and stored in a register

(vi) special types of register R(13) manages the stack, R14 holds return address and R15 points to next instructions

(vii) These continuous process allows the ARM processore to execute instructions quickly.

ARM memory map

The ARM processor uses a single 4GB memory space for both memory and I/O operations. This method is called memory mapped I/O, where I/O devices are treated as memory location.



In most ARM microcontrollers, this 4GB Memory is divided into three main sections.

1. I/O Registers (peripherals) (starting at 0x4000 000)

→ used for controlling devices like timers serial ports and ADCs

→ Each register has a fixed address decided by the chip manufacturer

→ The address and number of peripherals can vary from one chip to another chip

2. On-chip-SRAM (Static RAM) starting at 0x2000 0000

↳ used to store variables, stack and temporary data during program execution.

↳ size varies between different ARM chips

↳ sometimes it can both data and program

3. On-chip Flash ROM (Starting at 0x00000000)

→ used to store program code and constant data like lookup tables

→ size and location vary depending on the microcontroller.

CPSR Register

current program status Register

31	30	29	28	27	26	25	24	23 - 20	19 - 16	15 - 10	9	8	7	6	5	4	3	2	1	0
N	Z	C	V	Q	IT 1	IT 0	J	Reserved	GE[3:0]	IT[7:2]	E	A	I	F	T	M4	M3	M2	M1	M

1. Bit 31(N):- Becomes 1 if Result is negative

2. Bit 30(Z): set 1 if Result is zero.

3. Bit(29) C: Becomes 1 when a carry or borrow occurs

3. bit 28(V) : Becomes 1 when signed overflow happens.

4. bit 27(Q) : ~~set to 1 if signed~~ shows saturation occured in DSP (Digital ~~Singl~~ signal processing) operations.

5. bit 26(IT1) and bit 25(IT0) :- used for conditional execution in Thumb mode.

6. Bit 24(J) :- Enables Jazelle mode for running Java bytecode

7. Bit 23-20(Reserved) : Not used.

8. Bit 19-16 (GE[3:0] : Show greater than or equal results for 4 bytes in SIMD operation

9. Bit 15-10 (IT [7:2]) :- control conditional execution for Thumb instructions.

10. Bit 9 (E) chooses memory order 0 for Little Endian and 1 for Big Endian.

11. Bit 8 (A) :- Disables imprecise data aborts.

12. Bit 7 (I) :- Disables normal interrupt (IRQ)

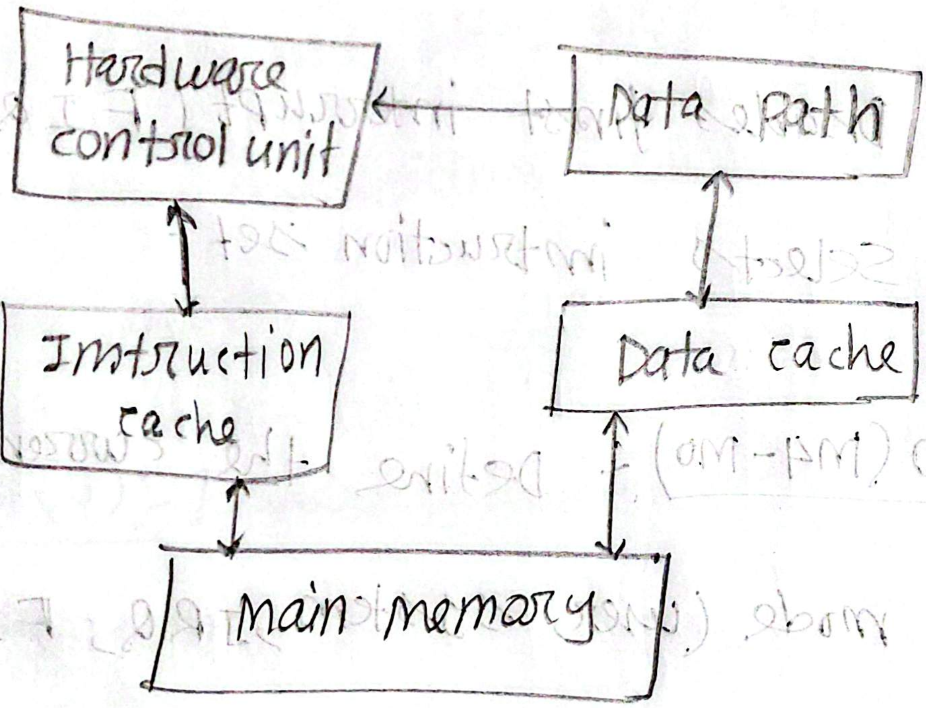
13. Bit 6 (F): Disables first interrupt (FIQ)

14. Bit 5 (T): Selects instruction set

15. Bit 4-0 (M4-M0) :- Define the current processor mode (user, system, IRQ, FIQ etc)

Q. Briefly describe some features of RISC

Ans The ARM microcontroller follow RISC (Reduced Instruction set computer) architecture, which focuses on simplicity, speed, and efficiency.



RISC Architecture

The main features are:

1. Fixed Instruction size:-

All instructions are of a fixed length (32bit) which makes decoding faster and simpler.

2. Large number of Registers:-

ARM provides many general-purpose registers to reduce memory access and increase execution speed.

3. Small instruction set:

ARM uses a limited set of simple instructions (around 50 types) making operations faster and easier to pipeline.

4. single-cycle Execution:-

Most ARM instructions are executed in one clock cycle.

Q. How ARM is programed and become ready to be installed into a system?

To program an ARM processor and make it Read to run, the following steps are followed:

1. Writing the program:

The assembly language program is written using a text editor or IDE such as Keil, which provide an editor, assembler, simulator and debugger.

The program is saved with an extension Like .asm, .s, .a.

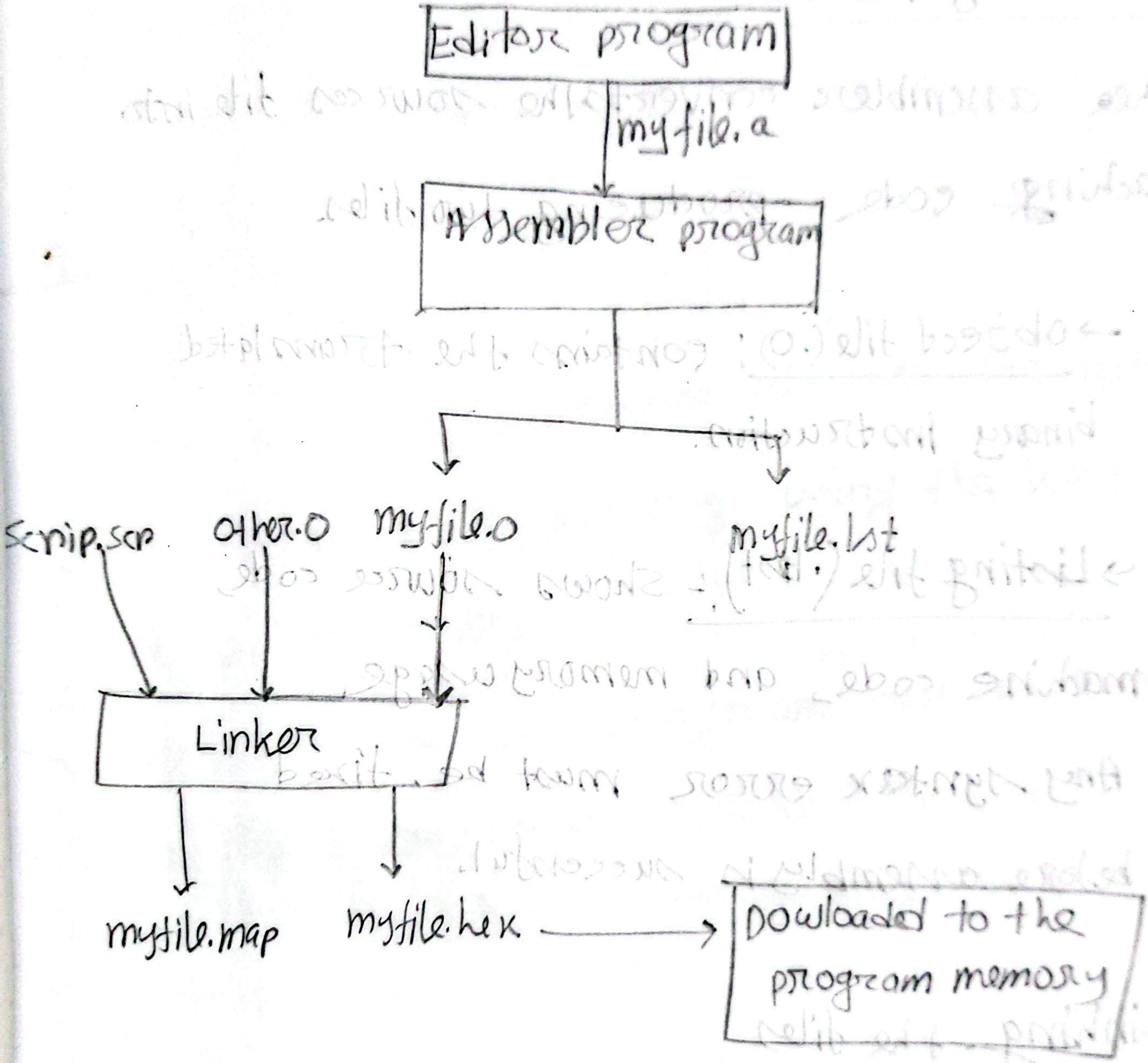


fig: steps to create a program and installed.

2 Assembling the code

The assembler converts the sources file into machine code producing two files

→ object file (.o): contains the translated binary instruction.

→ Listing file (.lst): shows source code, machine code, and memory usage.

Any syntax error must be fixed before assembly is successful.

3 Linking the files

The linker combines the object file and linker script to produce:

→ map file (.map): shows label address and memory layout

- memory image file (.hex): contains the final executable binary.

4. **loading into target system**

The .hex file is downloaded into the ARM processors. Flash memory using the keil IDE or other tools.

once loaded, the program can run directly on the ARM chip.

Q: Difference between microprocessors and microcontrollers

Feature	Microprocessor	microcontroller
main components	Only contains the CPU	contain CPU, RAM, ROM, I/O on a single chip
system	Needs external components (RAM, ROM) to function	It is complete self-contained computer on a chip
Application	used in complex system like computers and servers	used in simple tasks like washing machines, toys and embadded system
power consumption	Generally high power consumption	Generally low power consumption

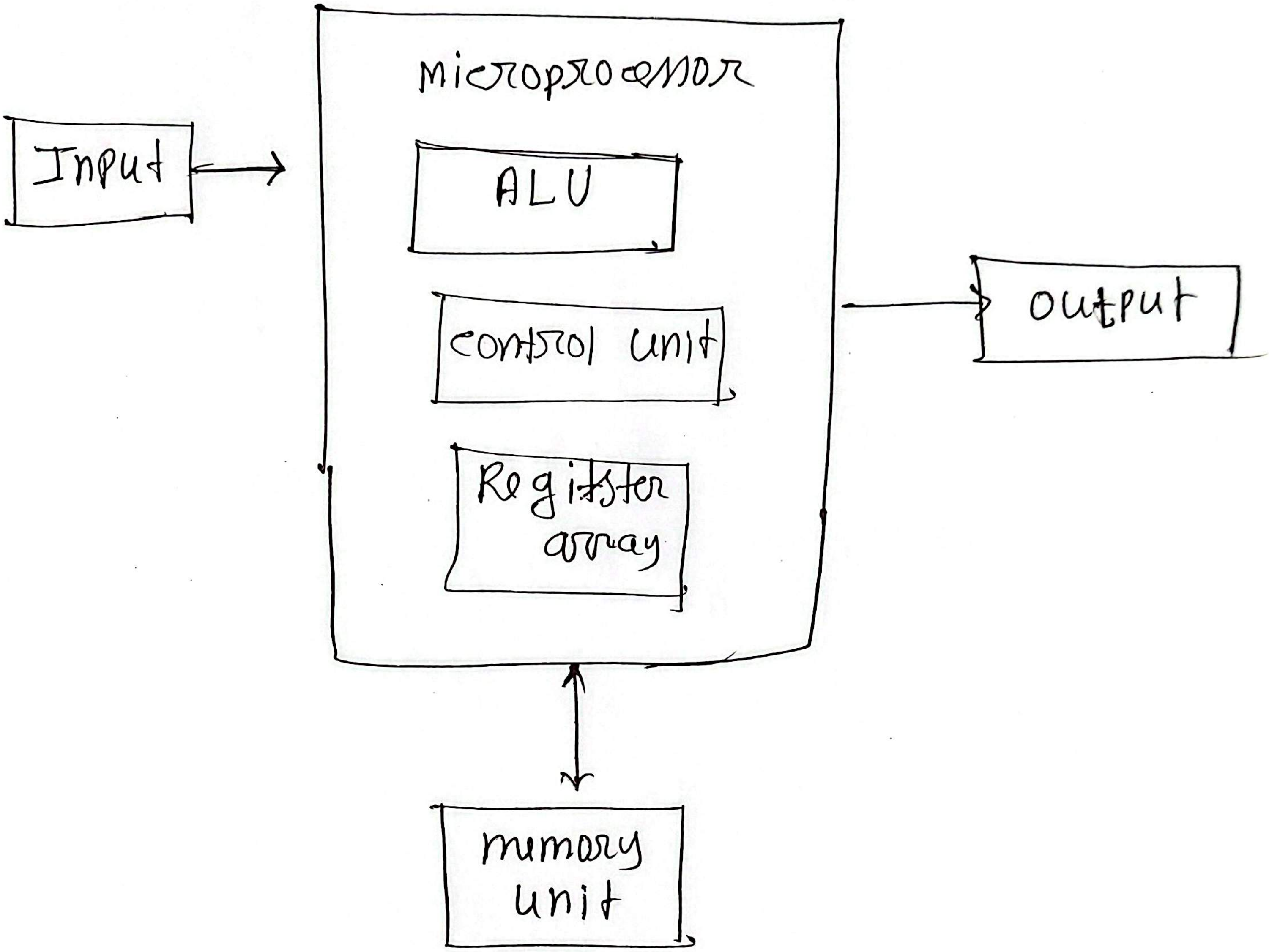


fig: microprocessor's block diagram.