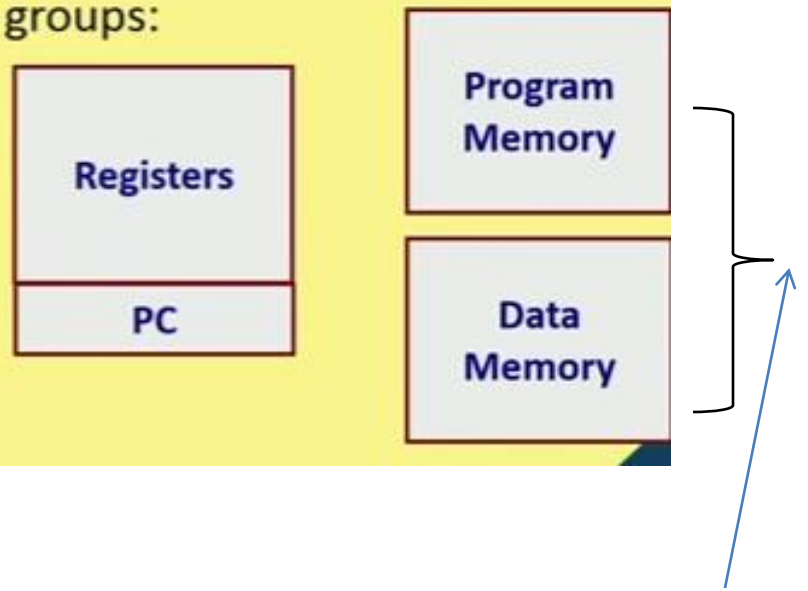
EEE1024: Fundamentals of Electrical and Electronics Engineering

Dr. Sanchit Khatavkar

INSTRUCTION SET of ARM Processor

The ARM INSTRUCTION SET

- ARM instruction can be categorized into three groups:
  - a) Data processing instructions
    - Operate on values in registers
  - b) Data transfer instructions
    - Move values between registers and memory
  - c) Control flow instructions
    - Change the value of the program counter (PC)



Harvard architecture

Data Processing (ALU) Instructions - 1

Arithmetic instructions:

```
ADD   r0,r1,r2    ; r0 = r1 + r2
ADC   r0,r1,r2    ; r0 = r1 + r2 + C   (C is carry bit)
SUB   r0,r1,r2    ; r0 = r1 - r2
SBC   r0,r1,r2    ; r0 = r1 - r2 + C - 1
RSB   r0,r1,r2    ; r0 = r2 - r1
RSC   r0,r1,r2    ; r0 = r2 - r1 + C - 1
```

Borrow

- **Register-register move instructions:**

```
MOV   r0,r2       ; r0 = r2
MVN   r0,r2       ; r0 = not r2
```

- MVN is the acronym for "*move negated*"
  - Each 1-bit in r2 clears the corresponding bit in r0.

- **Bit-wise logical instructions:**

```
AND   r0,r1,r2    ; r0 = r1 and r2
ORR   r0,r1,r2    ; r0 = r1 or  r2
EOR   r0,r1,r2    ; r0 = r1 xor r2
BIC   r0,r1,r2    ; r0 = r1 and not r2
```

- BIC is the acronym for "*bit clear*"
  - Each 1-bit in r2 clears the corresponding bit in r1.

No assignment of result in any register!

- **Comparison instructions:**

```
CMP   r1,r2       ; set cc on (r1 - r2)
CMN   r1,r2       ; set cc on (r1 + r2)
TST   r1,r2       ; set cc on (r1 and r2)
TEQ   r1,r2       ; set cc on (r1 xor r2)
```

- All these instructions affect the condition codes (N, Z, C, V) in the current program status register (CPSR).

Is r1 greater than r2?

TEQ → $0 \oplus 0 = 0$, $1 \oplus 1 = 0$

$0 \oplus 1 = 1$, $1 \oplus 0 = 1$

Data Processing (ALU) Instructions - 2

- **Shifted register operands:**
  - The second source operand may be shifted either by a constant number of bit positions, or by a register-specified number of bit positions.

```
ADD    r1,r2,r3,LSL #3    ; r1 = r2 + (r3 << 3)
ADD    r1,r2,r3,LSL r5    ; r1 = r2 + (r3 << r5)
```

- Various shift and rotate options:
  - **LSL**: logical shift left
  - **LSR**: logical shift right
  - **ROR**: rotate right
  - **RRX**: rotate right extended by 1 bit
  - **ASL**: arithmetic shift left
  - **ASR**: arithmetic shift right

- **Specifying immediate operands:**

```
ADD    r1,r2,#2           ; r1 = r2 + 2
SUB    r3,r3,#1           ; r3 = r3 - 1
```

- **Multiplication instruction**

```
MUL    r1,r2,r3           ; r1 = (r2 x r3)[31:0]
```

  - Only the least significant 32-bits are returned.
  - Immediate operands are not supported.

- **Multiply-accumulate instruction:**

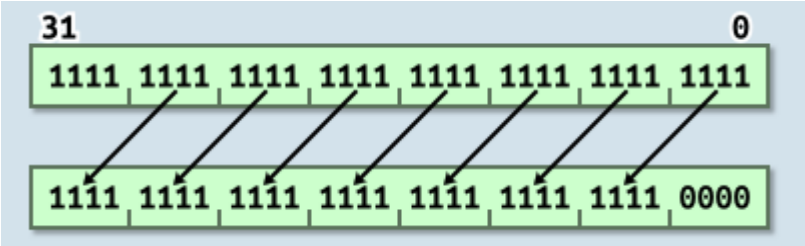
```
MLA    r1,r2,r3,r4        ; r1 = (r2 x r3 + r4)[31:0]
```

  - Required in digital signal processing (DSP) applications.

ARM – Barrel Shifting

•Barrel shifter is a hardware that allows multiple bit shifting in 1 cycle.

•It performs SHIFT and ROTATE operations in *ARM* processors - 5 types



1. Logic Shift Left (LSL),

2.LSR,

3. Arithmetic Shift Right (ASR),

4. ROR,

5. RRX

Data Transfer Instructions - 1

- ARM instruction set supports three types of data transfers:
  a) Single register loads and stores
     - Flexible, supports byte, half-word and word transfers
  b) Multiple register loads and stores
     - Less flexible, multiple words, higher transfer rate
  c) Single register-memory swap
     - Mainly for system use (for implementing locks)

Data Transfer Instructions - 2

- Before any data transfer, some register must be initialized with a memory address.

```
ADRL   r1,Table      ; r1 = memory address of Table
```

- Example:

```
LDR    r0,[r1]       ; r0 = mem[r1]
STR    r0,[r1]       ; mem[r1] = r0
```

Single register loads and stores

- **Multiple register loads and stores**
  - ARM supports instructions that transfer between several registers and memory.
  - Example:

```
LDMIA r1,{r3,r5,r6}      ; r3 = mem[r1]
                         ; r5 = mem[r1+4]
                         ; r6 = mem[r1+8]
```

  - For **LDMIB**, the addresses will be **r1+4**, **r1+8**, **and r1+12**.
  - The list of destination registers may contain any or all of r0 to r15.

Note: This is only for reference

Control Flow Instructions

- These instructions change the order of instruction execution.
  - Normal flow is sequential execution, where PC is incremented by 4 after executing every instruction.
- Types of conditional flow instructions:
  - Unconditional branch
  - Conditional branch
  - Branch and Link
  - Conditional execution

❑ Conditional execution instructions

- An example: `if (r2 != 10) r5 = r5 +10 – r3`
- Various instruction postfix supported for conditional execution:

Postfix	Condition	Postfix	Condition
CS	Carry set	CC	Carry clear
EQ	Equal (zero set)	NE	Not equal (zero clear)
VS	Overflow set	VC	Overflow clear
GT	Greater than	LT	Less than
GE	Greater than or equal	LE	Less than or equal
PL	Plus (positive)	MI	Minus (negative)
HI	Higher than	LO	Lower than (i.e. CC)
HS	Higher or same (i.e. CS)	LS	Lower or same

❑ Branch and link instruction

- Branch conditions that are supported:
  - B, BAL — Unconditional branch
  - BEQ, BNE — Equal or not equal to zero
  - BPL, PMI — Result positive or negative
  - BCC, BCS — Carry set or clear
  - BVC, BVS — Overflow set or clear
  - BGT, BGE — Greater than, greater or equal
  - BLT, BLE — Less than, less or equal

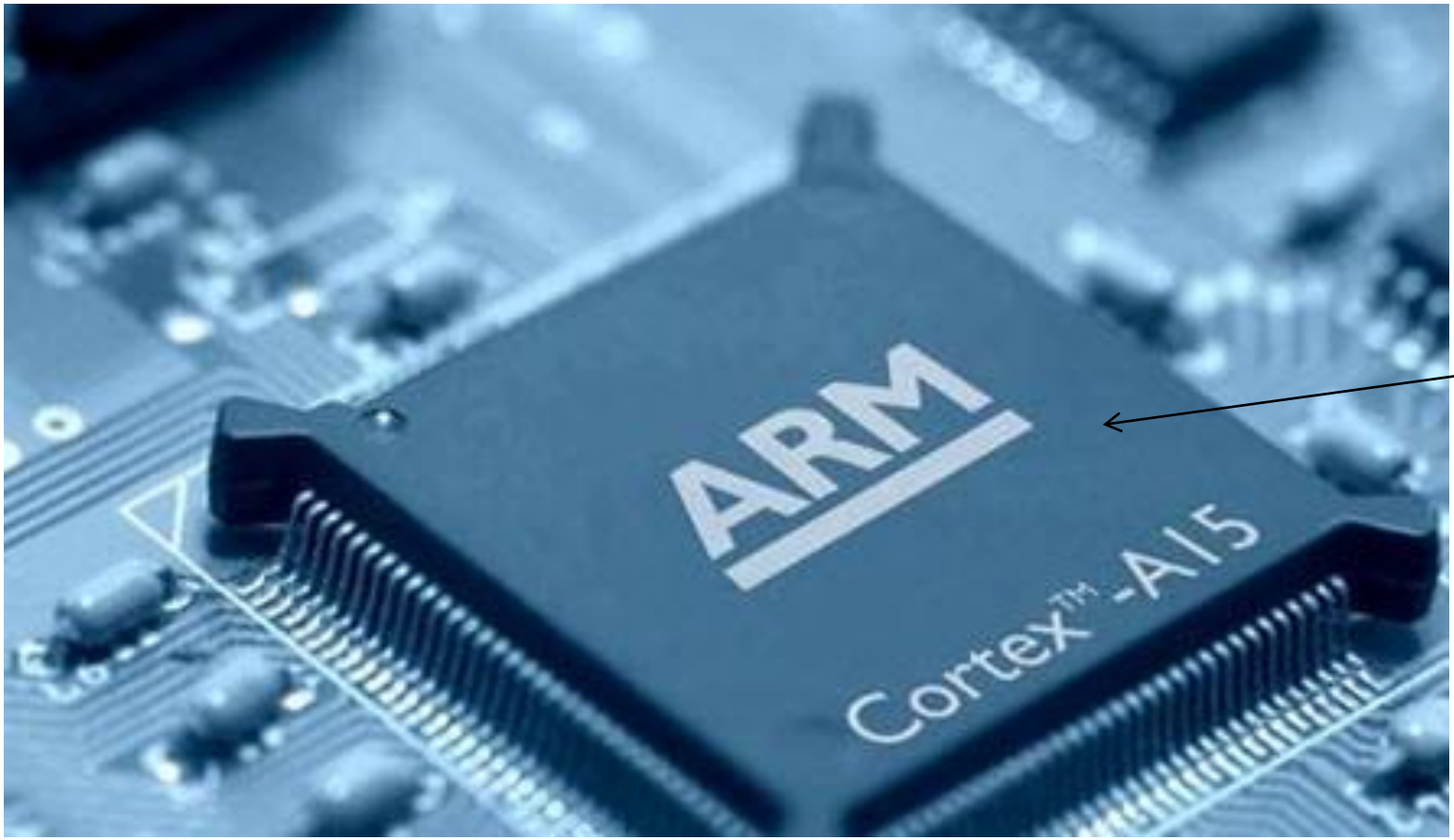
Note: This is only for reference

The 8051 Microcontroller

Microprocessor Vs Microcontroller

ALU	Control Unit	Registers

Processor	Memory (RAM,ROM,EPROM, etc)	Serial COM Ports	Peripherals (Timers, Counters, etc.)



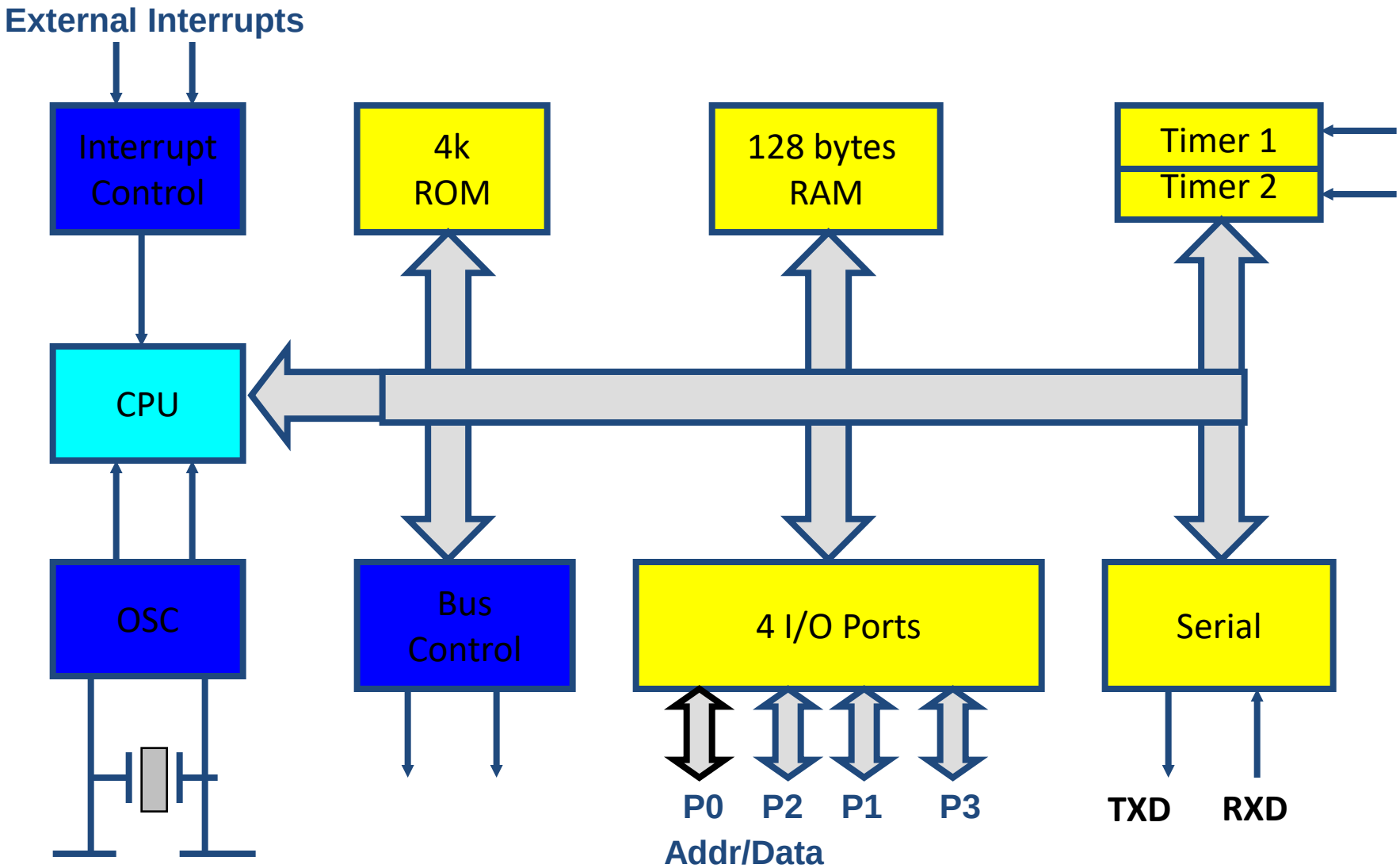
CHIP

	Microprocessor	Microcontroller
Application	It used where **intensive processing** is required. It is used in personal computers, laptops, mobiles, video games, etc.	It used where the **task is fixed and predefined**. It is used in the washing machine, alarm, etc.
Structure	It has only the CPU in the chip. Other devices like I/O port, memory, timer is connected externally. The structure of the microprocessor is flexible. Users can decide the amount of memory, the number of I/O port and other peripheral devices.	CPU, Memory, I/O port and all other devices are connected on the single chip. The structure is fixed. Once it is designed the user cannot change the peripheral devices.
Clock speed	The **clock speed** of the microprocessor **is high**. It is in terms of the GHz. It ranges between 1 GHz to 4 GHz.	The **clock speed** of the microcontroller **is less**. It is in terms of the MHz. it ranges between 1 MHz to 300 MHz.
RAM	The volatile memory (RAM) for the microprocessor is in the range of the 512 MB to 32 GB.	The volatile memory (RAM) for the microcontroller is in the range of 2 KB to 256 KB.
ROM	The hard disk (ROM) for the microprocessor is in the range of the 128 GB to 2 TB.	The hard drive or flash memory (ROM) is in the range of the 32 KB to 2 MB.
Peripheral interface	The common peripheral interface for the microprocessor is USB, UART, and high-speed Ethernet.	The common peripheral interface for the microcontroller is I2C, SPI, and UART.
Programming	The program for the microprocessor can be changed for different applications. The programming of the microprocessor is difficult compared to the microcontroller.	The program for the microcontroller is fixed once it is designed.
Bit size	It is available in **32-Bit and 64-bit.**	It is available in **8-bit, 16-bit, and 36-bit.**
Cost	The cost of the microprocessor is high compared to the microcontroller.	It is cheaper.
Power consumption	The power consumption for the microprocessor is high.	The power consumption for the microcontroller is less.
Size	The overall size of the system is large.	The overall size of the system is small.

8051 Basic Component

- 4K bytes internal ROM
- 128 bytes internal RAM
- Four 8-bit I/O ports (P0 - P3).
- Two 16-bit timers/counters
- One serial interface

Block Diagram



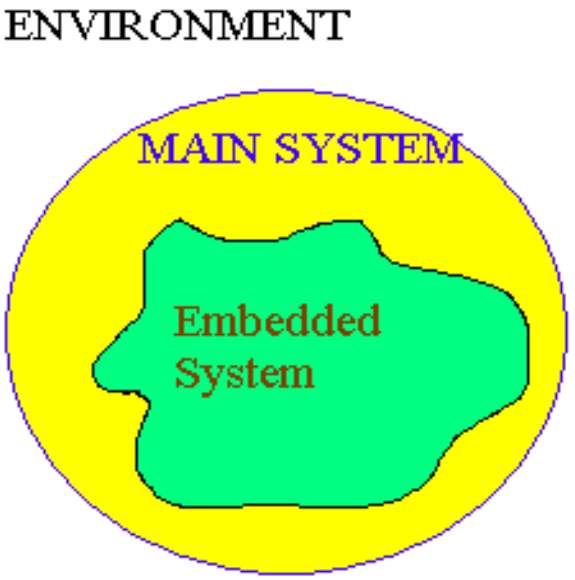
Open Sound Control (**OSC**)

Other 8051 features

- only 1 On chip oscillator (external crystal)
- 6 interrupt sources (2 external , 3 internal, Reset)
- 64K external code (program) memory(only read)PSEN
- 64K external data memory(can be read and write) by RD,WR
- Code memory is selectable by EA (internal or external)
- We may have External memory as data and code

Embedded System
(8051 Application)

- What is Embedded System?
  - An embedded system is closely integrated with the main system
  - It may not interact directly with the environment
  - For example – A microcomputer in a car ignition control



❖ An embedded product uses a microprocessor or microcontroller to do one task only

❖ There is only one application software that is typically burned into ROM

Examples of Embedded Systems

- Keyboard
- Printer
- video game player
- MP3 music players
- Embedded memories to keep configuration information
- Mobile phone units
- Domestic (home) appliances
- Data switches
- Automotive controls

Comparison of the 8051 Family Members

- ROM type
  - 8031 no ROM
  - 80xx mask ROM
  - 87xx EPROM
  - 89xx Flash EEPROM
- 89xx
  - 8951
  - 8952
  - 8953
  - 8955
  - 898252
  - 891051
  - 892051
- Example (AT89C51,AT89LV51,AT89S51)
  - AT= ATMEL(Manufacture)
  - C = CMOS technology
  - LV= Low Power(3.0v)

Acknowledgements

1. https://www.watelectronics.com/arm-processor-architecture-working/
2. http://www.davespace.co.uk/arm/introduction-to-arm/barrel-shifter.html
3. https://developer.arm.com/documentation/dui0471/i/key-features-of-arm-architecture-versions/thumb-2-technology
4. https://www.arm.com/why-arm/technologies/dsp
5. https://nptel.ac.in/