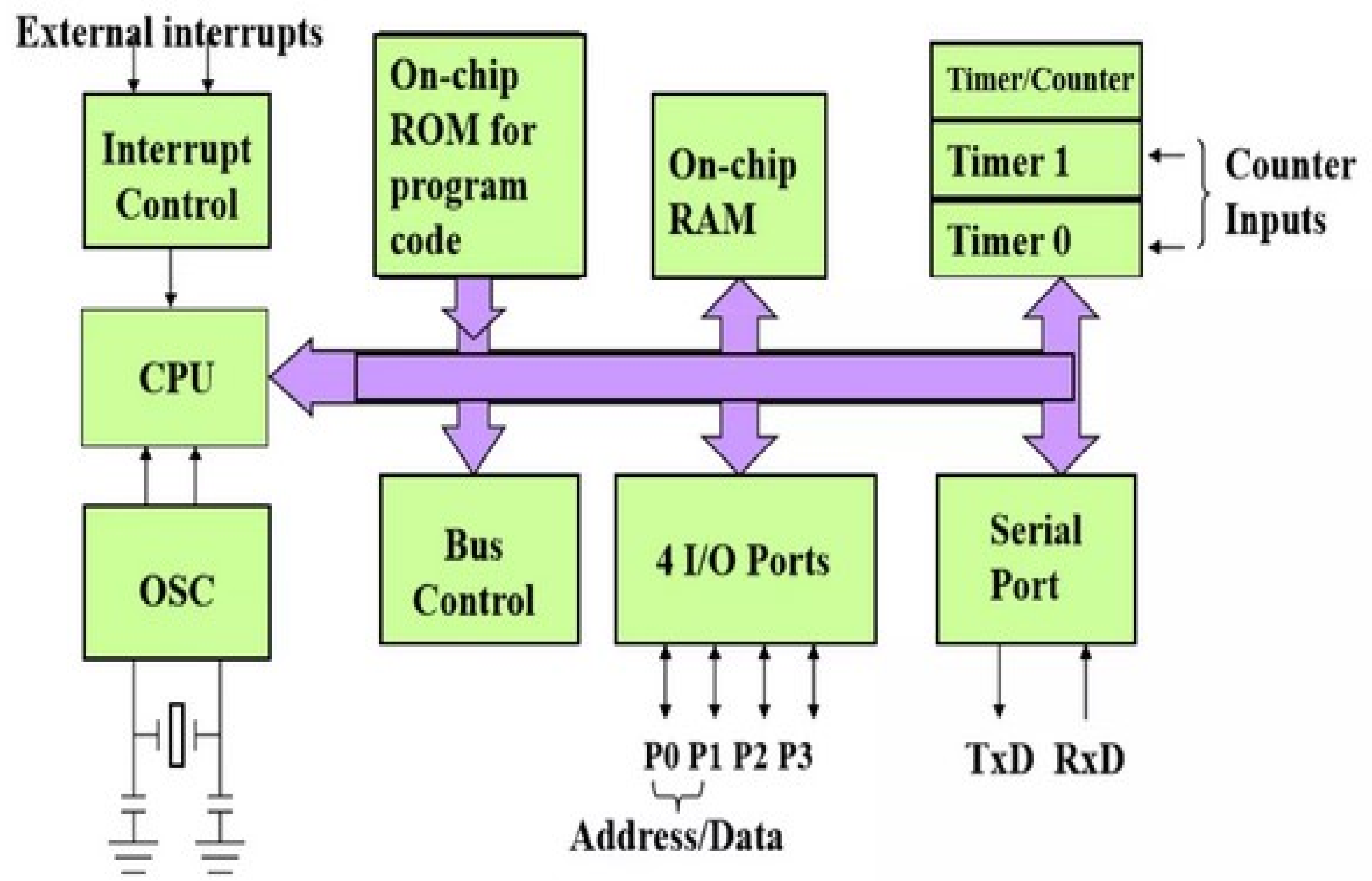
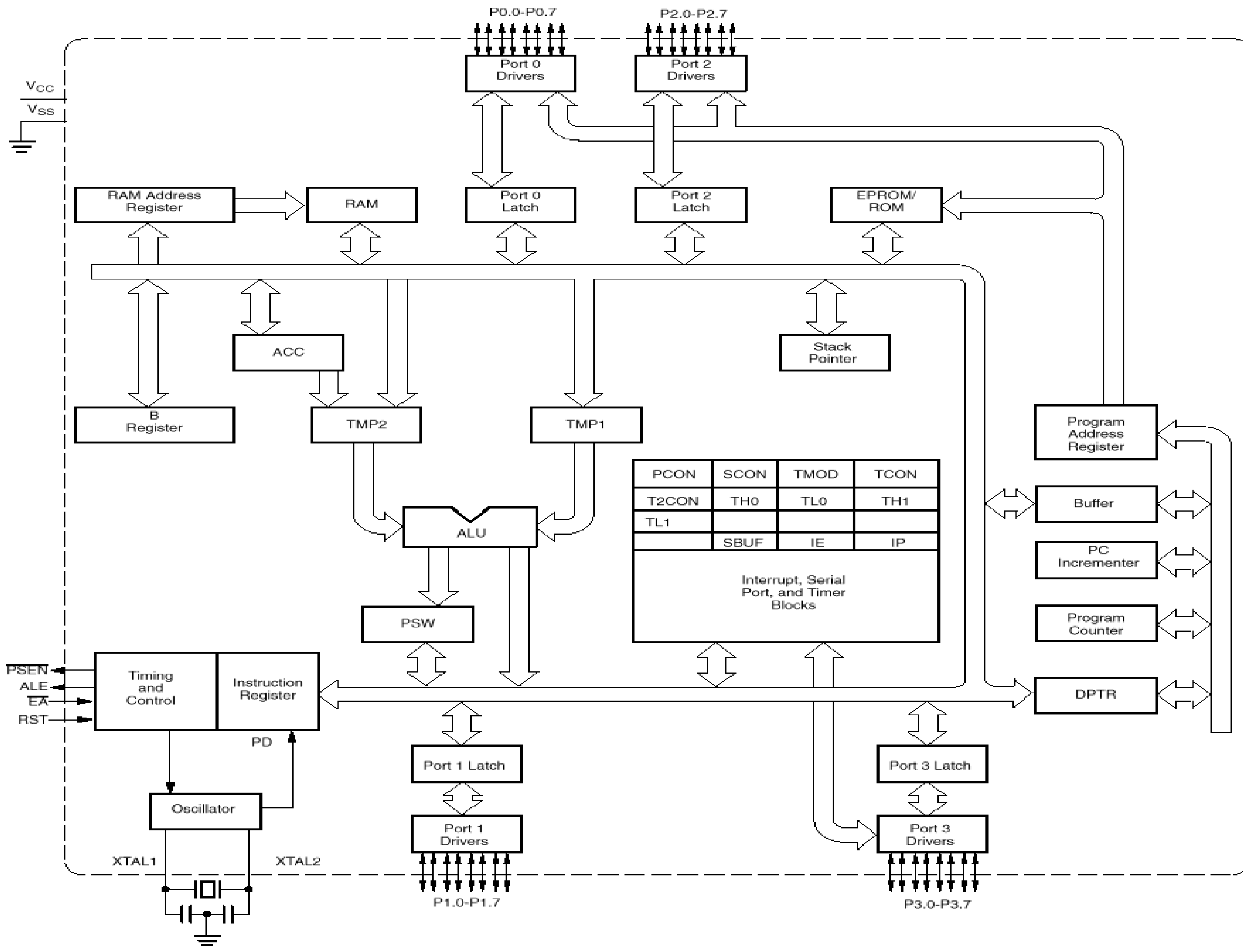
Module-1-Microcontrollers – Slide set-2

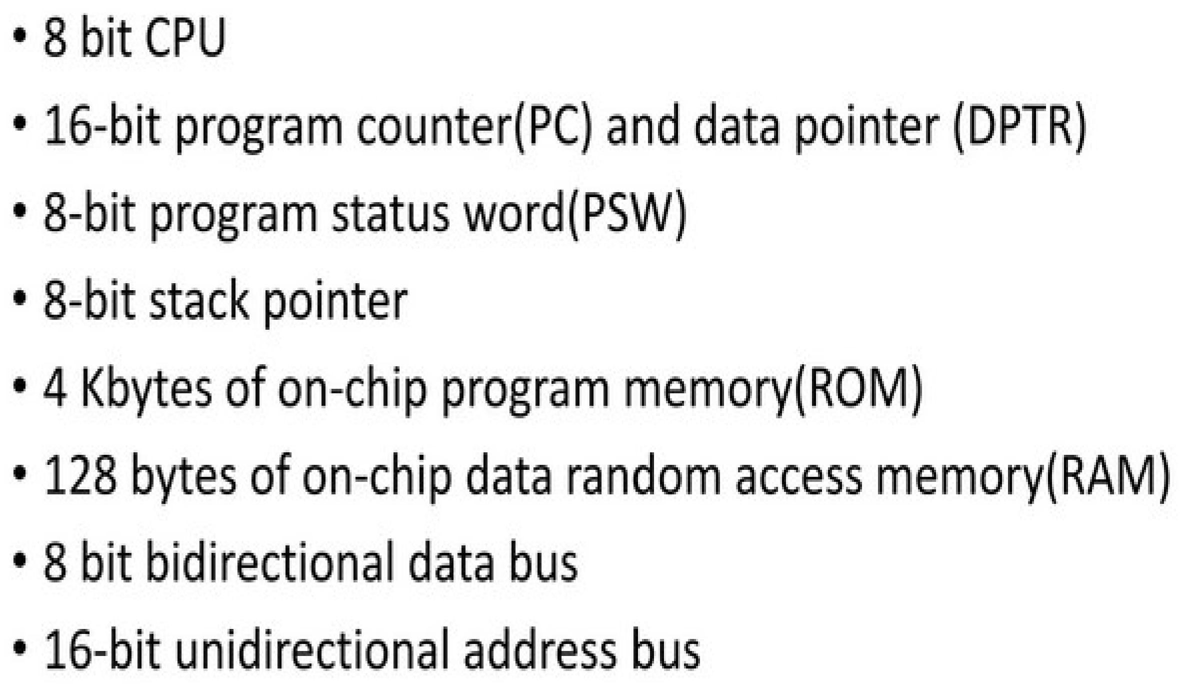
8051 Block Diagram



8051 Internal Block Diagram



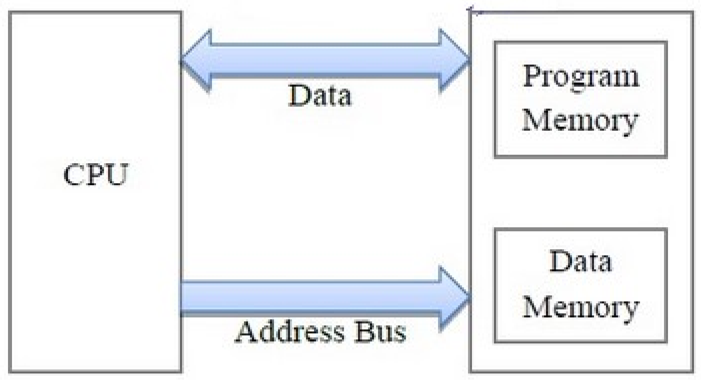
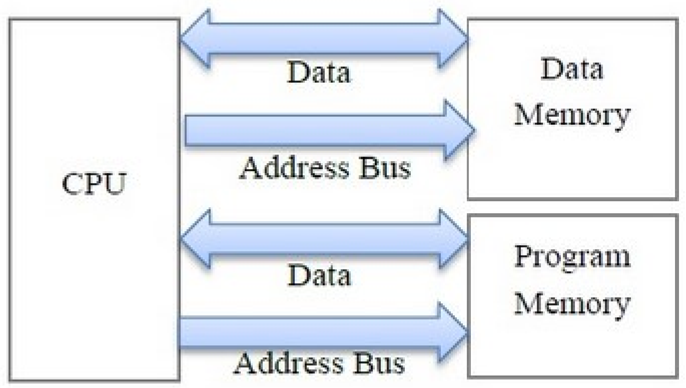
Features of 8051



- 8 bit CPU
- 16-bit program counter(PC) and data pointer (DPTR)
- 8-bit program status word(PSW)
- 8-bit stack pointer
- 4 Kbytes of on-chip program memory(ROM)
- 128 bytes of on-chip data random access memory(RAM)
- 8 bit bidirectional data bus
- 16-bit unidirectional address bus

Features of 8051

- 64Kbytes of program memory address space
- 64Kbytes of data memory address space
- Two 16 bit timers/counters
- 16 bit address bus multiplexed with port 0 and port 2
- 32 bit bidirectional I/O lines can be either used as 4 8-bit ports
- On-chip clock oscillator
- Control Registers
- Serial data receiver/transmitter
- Interrupt source
- Special features like USART, ADC, etc
- It is a CISC based Microcontroller with Harvard Architecture (separate program and data memory).

Von-Neumann (Princeton architecture)	**Harvard architecture**
	 CPU Data Address Bus Data Memory Data Address Bus Program Memory

ALU (Arithmetic Logic Unit)

- it is 8 bit ALU. Part of CPU
- Performs arithmetic, logical and bitwise operations
- Used in manipulating data (either 8 bit or 1 bit)
- Individual bit of any register can be set, reset, cleared, complimented with the help of logical computation,.
- Several registers are connected to it.

Instruction decoder and control

- When an instruction is fetched – it is loaded in the instruction register
- Decoder decodes the instruction and establishes the sequence of events to follow
- Instruction cycle: Instruction cycle is defined as the time required for completing the execution of an instruction

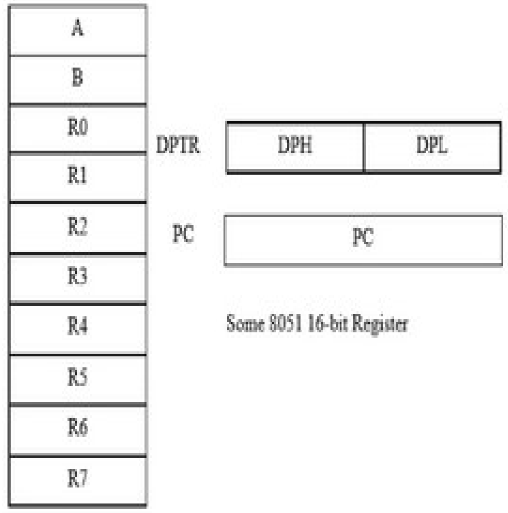
Accumulator or A register

- The Accumulator or Register A is the 8 bit register which is most important and most used 8051 Microcontroller SFRs.
- It is located at the address E0H in the SFR memory space.
- The Accumulator is used to hold or store the data for almost all the ALU Operations like:
- Arithmetic Operations like Addition, Subtraction, Multiplication etc.
- Logical Operations like AND, OR, NOT etc.
- Data Transfer Operations (between 8051 and External Memory)
- register is used to accumulate (or store) the result of all Arithmetic and most of the Logical Operations.
- It can hold an 8-bit (1-byte) value
- More than half of the 8051s 255 instructions manipulate or use the accumulator in some way.

Instruction Register (IR):

- store instructions
- When one instruction is completed, next instruction is fetched in memory for processing.

Registers

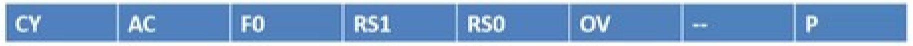


Some 8-bitt Registers of the 8051

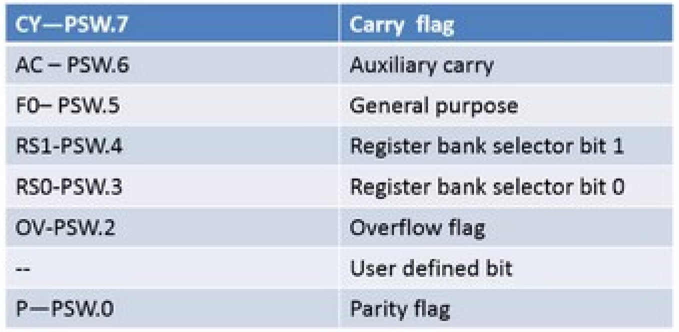
Some 8051 16-bit Register

Program status word(D0H)

- PSW or Program Status Word Register is also called as Flag Register and is one of the important SFRs.
- The PSW Register consists of Flag Bits, which help the programmer in checking the condition of the result and also make decisions.
- Flags are 1-bit storage elements that store and indicate the nature of the result that is generated by execution of certain instructions. The following image shows the contents of the PSW Register.



CY	AC	F0	RS1	RS0	OV	--	P



CY—PSW.7	Carry flag
AC – PSW.6	Auxiliary carry
F0– PSW.5	General purpose
RS1-PSW.4	Register bank selector bit 1
RS0-PSW.3	Register bank selector bit 0
OV-PSW.2	Overflow flag
--	User defined bit
P—PSW.0	Parity flag

Stack Pointer(81H)

- SP or Stack Pointer Contains data item on the top of stack and it indicates the next data to be accessed.
- The Stack Pointer is an 8-bit register and upon reset, the Stack Pointer is initialized with 07H.
- When writing a new data byte(storing) into the stack, the SP (Stack Pointer) is automatically incremented by 1 and the new data is written at an address SP+1.

Data Pointer

- The Data Pointer is a 16-bit Register and is physically the combination of DPL (Data Pointer Low) and DPH (Data Pointer High) SFRs.
- Therefore Data Pointer can be used as a single 16-bit register (as DPTR) or two 8-bit registers (as DPL and DPH).

Program Counter

- Its 16 bit (2 byte) register
- Specifies the address of next instruction to be executed and is incremented each time an instruction is executed
- After reset – PC will be set to 0000H and after execution of one instruction, PC is incremented automatically to point to the address of the next instruction to be executed.

8051 Memory

- The data width is 8 bits
- Registers are 8 bits
- Addresses are 8 bits
  - i.e. addresses for only 256 bytes!
  - PC is 16 bits (up to 64K program memory)
  - DPTR is 16 bits (for external data - up to 64K)

Program Memory

- Program and Data memory are separate
- Can be internal and/or external
    - 20K internal flash for the Atmel controller
- Read-only
    - Instructions
    - Constant data

Data Memory

- External Data - **`xdata`**
  - Resides off-chip
  - Accessed using the DPTR and MOVX instruction
  - We will not use **`xdata`**
  - **We will use the SMALL memory model**
    - all data is on-chip
    - limited to only ~128 bytes of data!

- Internal data memory contains all the processor state
  - Lower 128 bytes: registers, general data
  - Upper 128 bytes:
    - indirectly addressed: 128 bytes, used for the stack (small!)
    - directly addressed: 128 bytes for "special" functions

Internal Data Memory

- Internal data memory contains all the processor state
  - Lower 128 bytes: registers, general data
  - Upper 128 bytes:
    - indirectly addressed: 128 bytes, used for the stack (small!)
    - directly addressed: 128 bytes for "special" functions

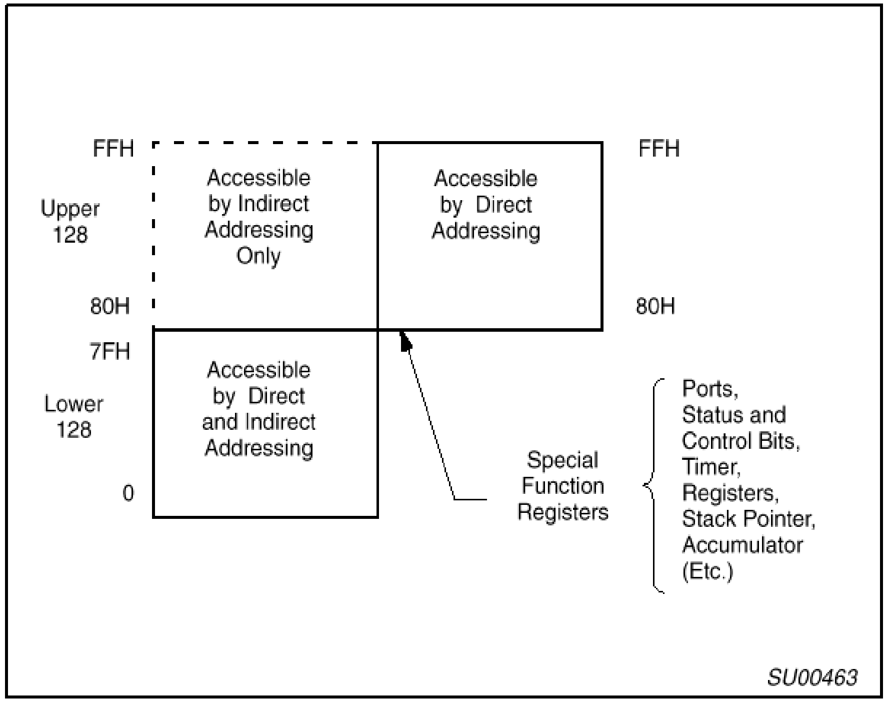

FFH
Upper 128
80H
7FH
Lower 128
0
Accessible by Indirect Addressing Only
Accessible by Direct Addressing
FFH
80H
Accessible by Direct and Indirect Addressing
Special Function Registers
Ports, Status and Control Bits, Timer, Registers, Stack Pointer, Accumulator (Etc.)
SU00463

Figure 6. Internal Data Memory

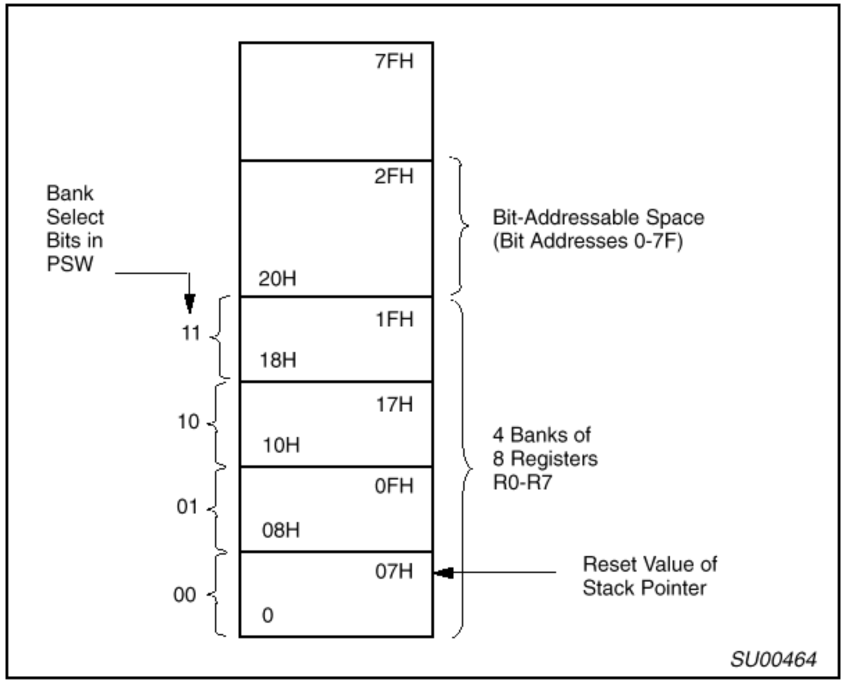

7FH
2FH
20H
1FH
18H
17H
10H
0FH
08H
07H
0
Bank Select Bits in PSW
11
10
01
00
Bit-Addressable Space (Bit Addresses 0-7F)
4 Banks of 8 Registers R0-R7
Reset Value of Stack Pointer
SU00464

Figure 7. Lower 128 Bytes of Internal RAM

Lower 128 bytes

- Register banks, bit addressable data, general data
  - you can address any register!

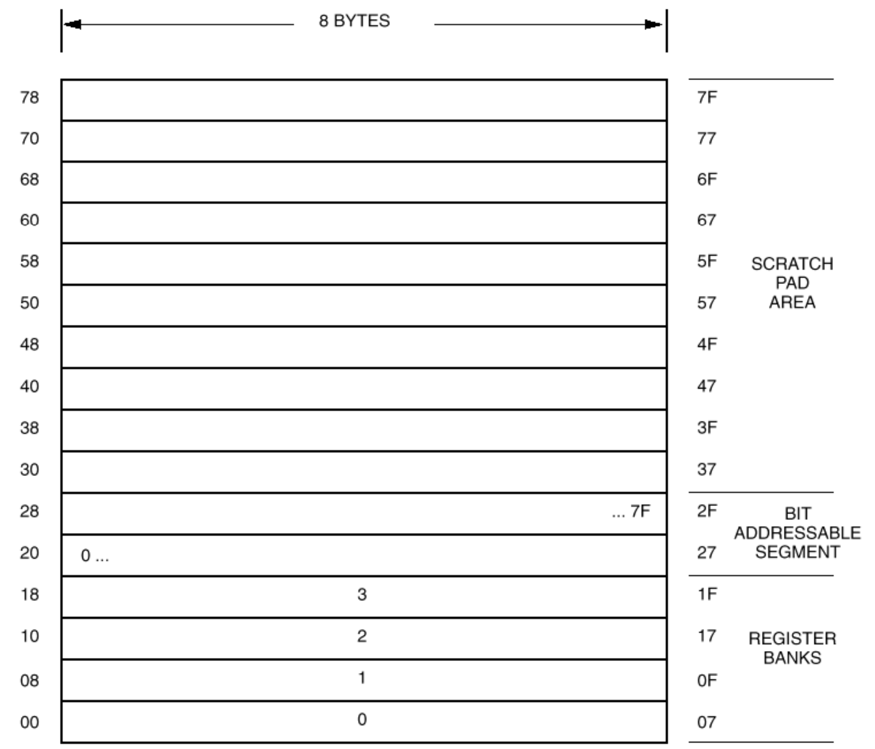


Figure 3. 128 Bytes of RAM Direct and Indirect Addressable

Ports

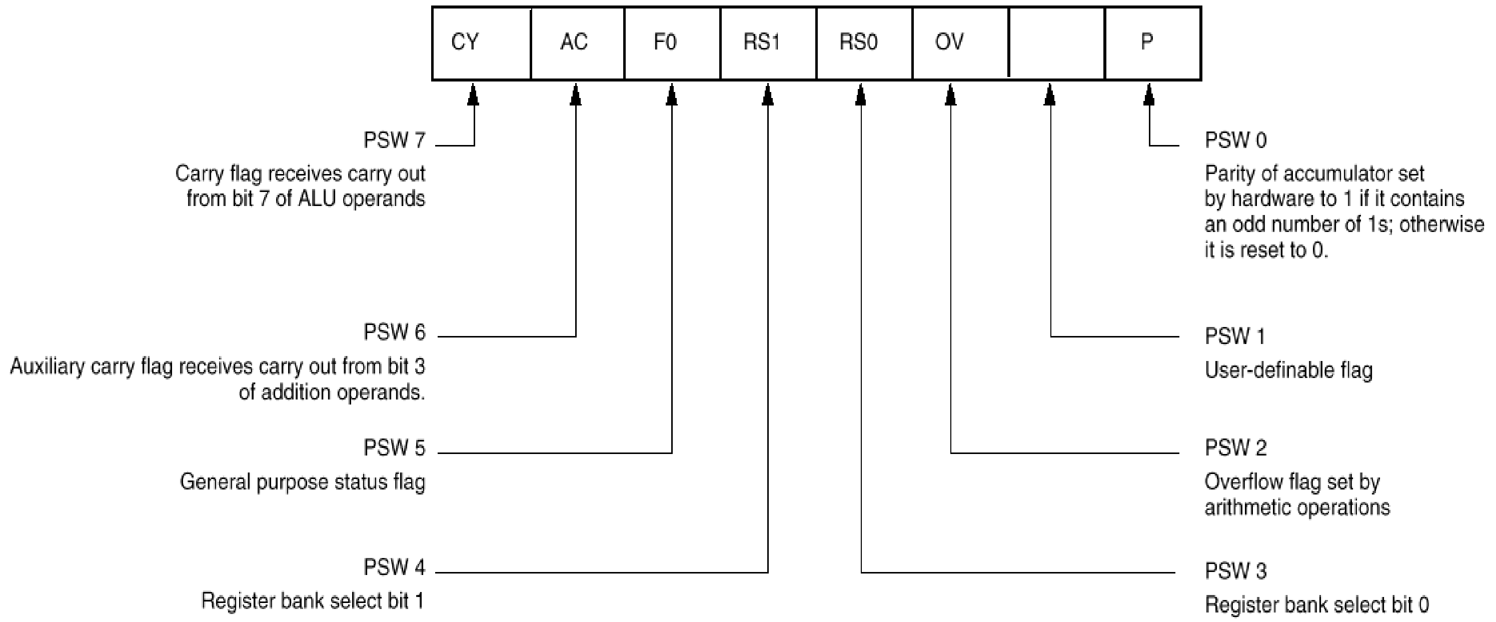
- Port 0 - external memory access
  - low address byte/data
- Port 2 - external memory access
  - high address byte
- Port 1 - general purpose I/O
  - pins 0, 1 for timer/counter 2
- Port 3 - Special features
  - 0 - RxD: serial input
  - 1 - TxD: serial output
  - 2 - INT0: external interrupt
  - 3 - INT1: external interrupt
  - 4 - T0: timer/counter 0 external input
  - 5 - T1: timer/counter 1 external input
  - 6 - WR: external data memory write strobe
  - 7 - RD: external data memory read strobe

Ports

- Port 0 - true bi-directional
- Port 1-3 - have internal pullups that will source current
- Output pins:
  - Just write 0/1 to the bit/byte
- Input pins:
  - Output latch must have a 1 (reset state)
    - Turns off the pulldown
    - pullup must be pulled down by external driver
  - Just read the bit/byte

Program Status Word

- Register set select
- Status bits



CY
AC
F0
RS1
RS0
OV
P
PSW 7
Carry flag receives carry out from bit 7 of ALU operands
PSW 6
Auxiliary carry flag receives carry out from bit 3 of addition operands.
PSW 5
General purpose status flag
PSW 4
Register bank select bit 1
PSW 0
Parity of accumulator set by hardware to 1 if it contains an odd number of 1s; otherwise it is reset to 0.
PSW 1
User-definable flag
PSW 2
Overflow flag set by arithmetic operations
PSW 3
Register bank select bit 0

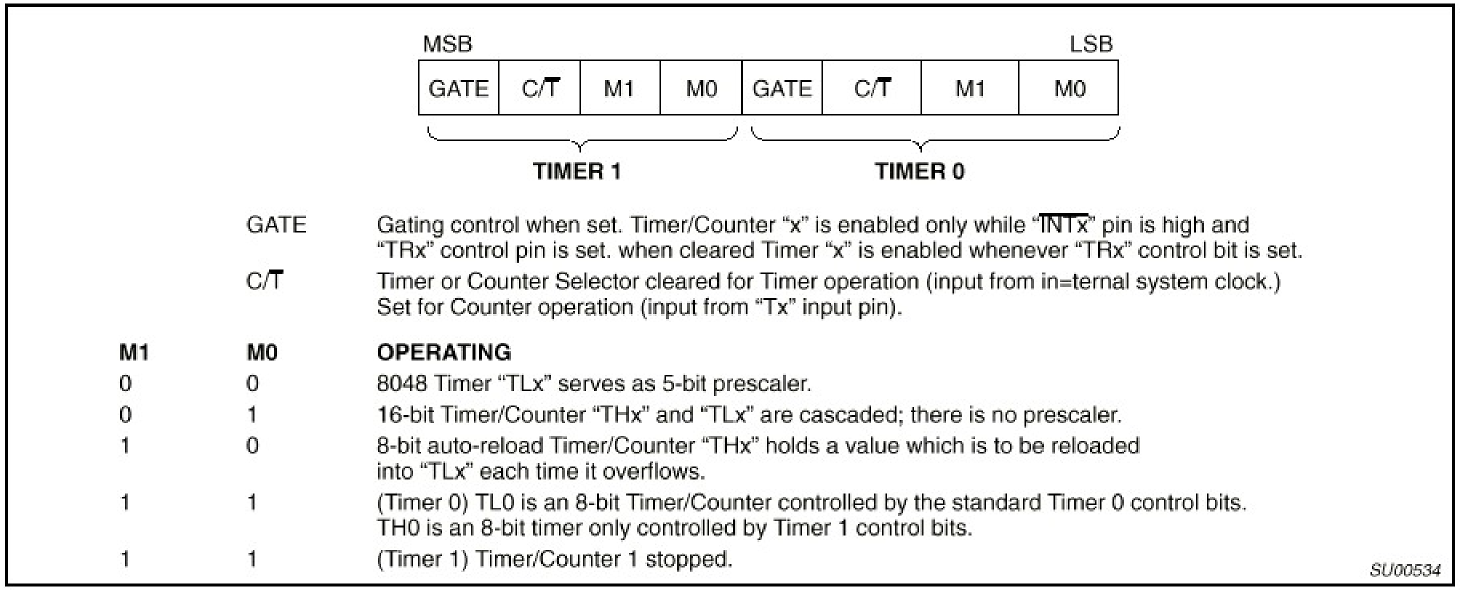
Figure 10. PSW (Program Status Word) Register in 80C51 Devices

Timers

- Base 8051 has 2 timers
  - we have 3 in the Atmel 89C55
- Timer mode
  - Increments every machine cycle (12 clock cycles)
- Counter mode
  - Increments when T0/T1 go from 1 - 0 (external signal)
- Access timer value directly
- Timer can cause an interrupt
- Timer 1 can be used to provide programmable baud rate for serial communications
- Timer/Counter operation
  - Mode control register (TMOD)
  - Control register (TCON)

Mode Control Register (TMOD)

- Modes 0-3
- GATE - allows external pin to enable timer (e.g. external pulse)
  - 0: INT pin not used
  - 1: counter enabled by INT pin (port 3.2, 3.3)
- C/T - indicates timer or counter mode



MSB — LSB

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
TIMER 1				TIMER 0			

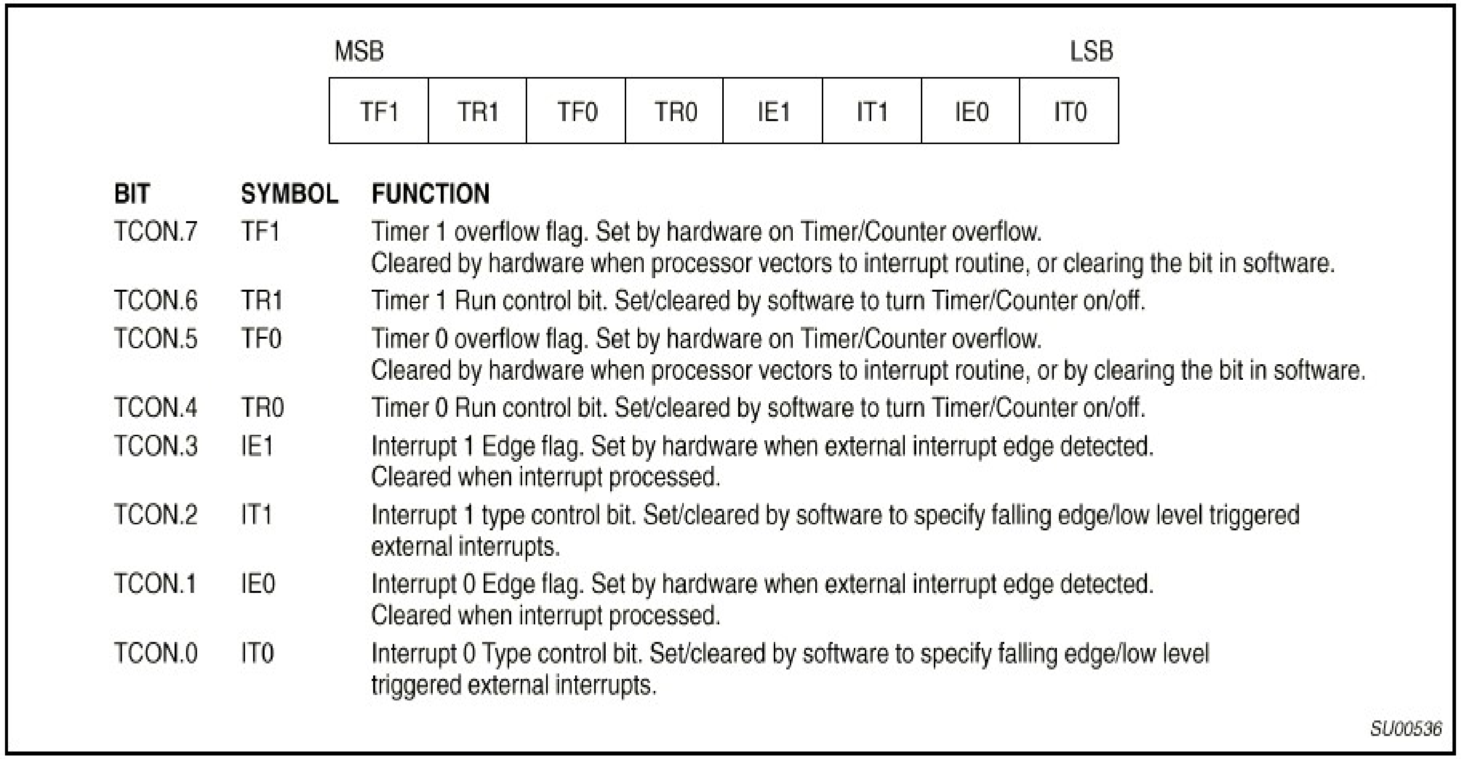
	GATE	Gating control when set. Timer/Counter “x” is enabled only while “$\overline{\text{INTx}}$” pin is high and “TRx” control pin is set. when cleared Timer “x” is enabled whenever “TRx” control bit is set.
	C/$\overline{T}$	Timer or Counter Selector cleared for Timer operation (input from in=ternal system clock.) Set for Counter operation (input from “Tx” input pin).
M1	**M0**	**OPERATING**
0	0	8048 Timer “TLx” serves as 5-bit prescaler.
0	1	16-bit Timer/Counter “THx” and “TLx” are cascaded; there is no prescaler.
1	0	8-bit auto-reload Timer/Counter “THx” holds a value which is to be reloaded into “TLx” each time it overflows.
1	1	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	(Timer 1) Timer/Counter 1 stopped.

SU00534

Figure 6. Timer/Counter Mode Control (TMOD) Register

Timer/Counter Control Register (TCON)

- TR - enable timer/counter
- TF - overflow flag: can cause interrupt
- IE/IT - external interrupts and type control
  - not related to the timer/counter



MSB LSB

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

BIT	SYMBOL	FUNCTION
TCON.7	TF1	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine, or clearing the bit in software.
TCON.6	TR1	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter on/off.
TCON.5	TF0	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine, or by clearing the bit in software.
TCON.4	TR0	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter on/off.
TCON.3	IE1	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TCON.2	IT1	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
TCON.1	IE0	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TCON.0	IT0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

SU00536

Figure 8. Timer/Counter Control (TCON) Register

Timer/Counter Mode 0

- Mode 1 same as Mode 0, but uses all 16 bits

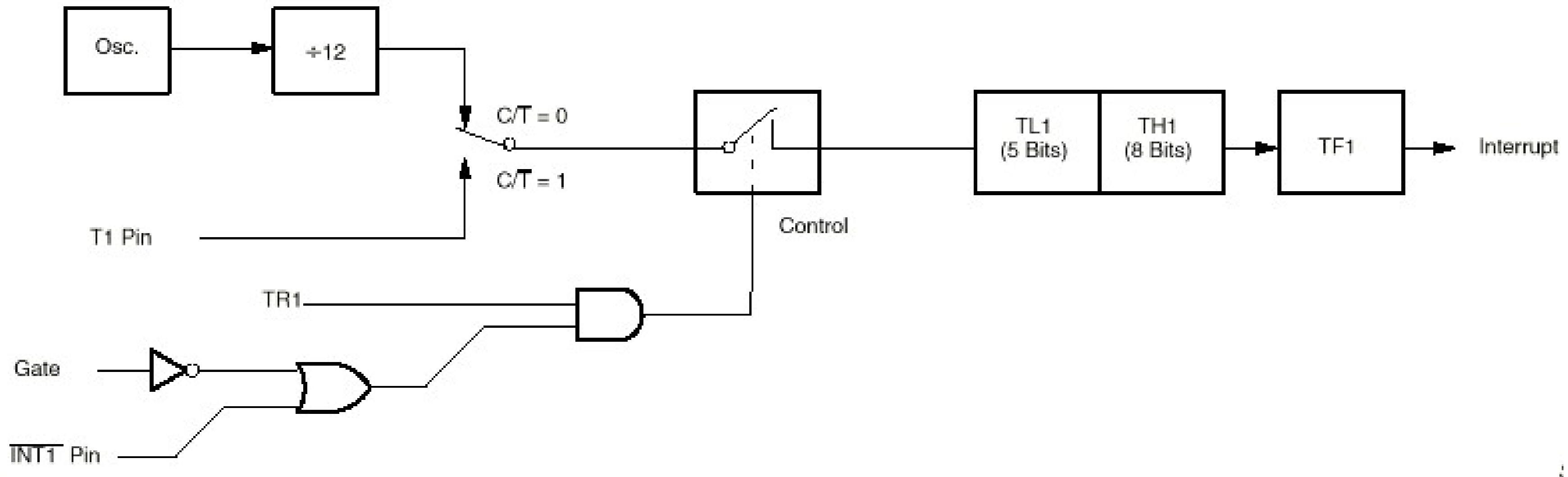


Figure 7. Timer/Counter Mode 0: 13-Bit Counter

Timer/Counter Mode 2

⌘ 8-bit counter, auto-reload on overflow

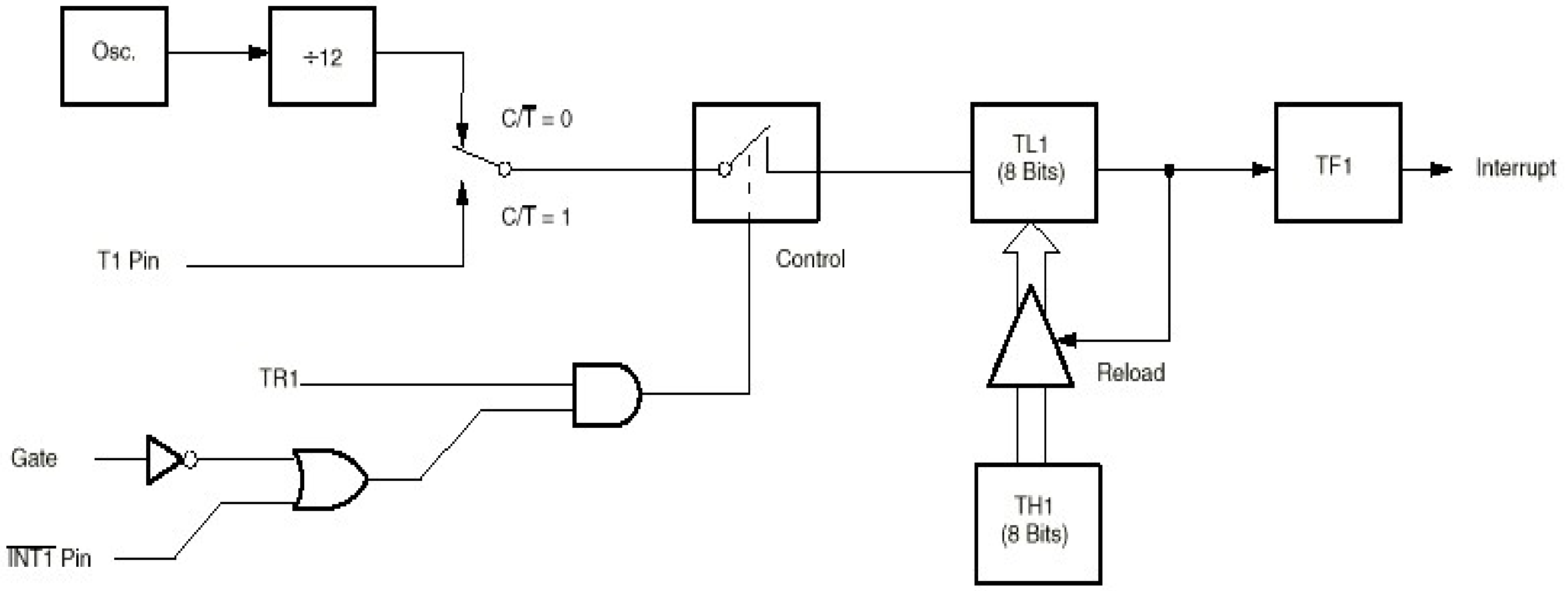


Figure 9. Timer/Counter Mode 2: 8-Bit Auto-Load

Timer/Counter Mode 3

- Applies to Timer/Counter 0
- Gives an extra timer

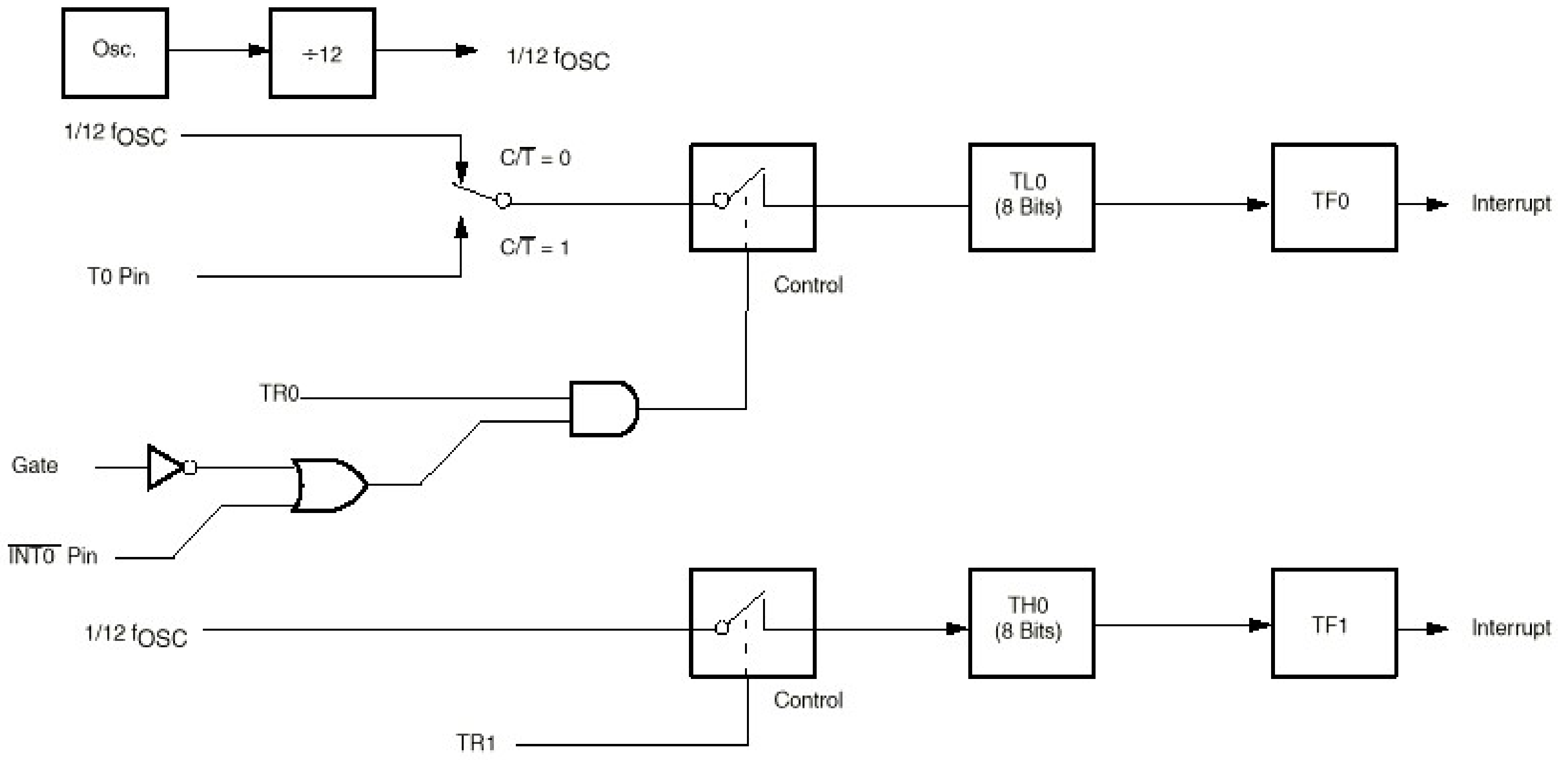
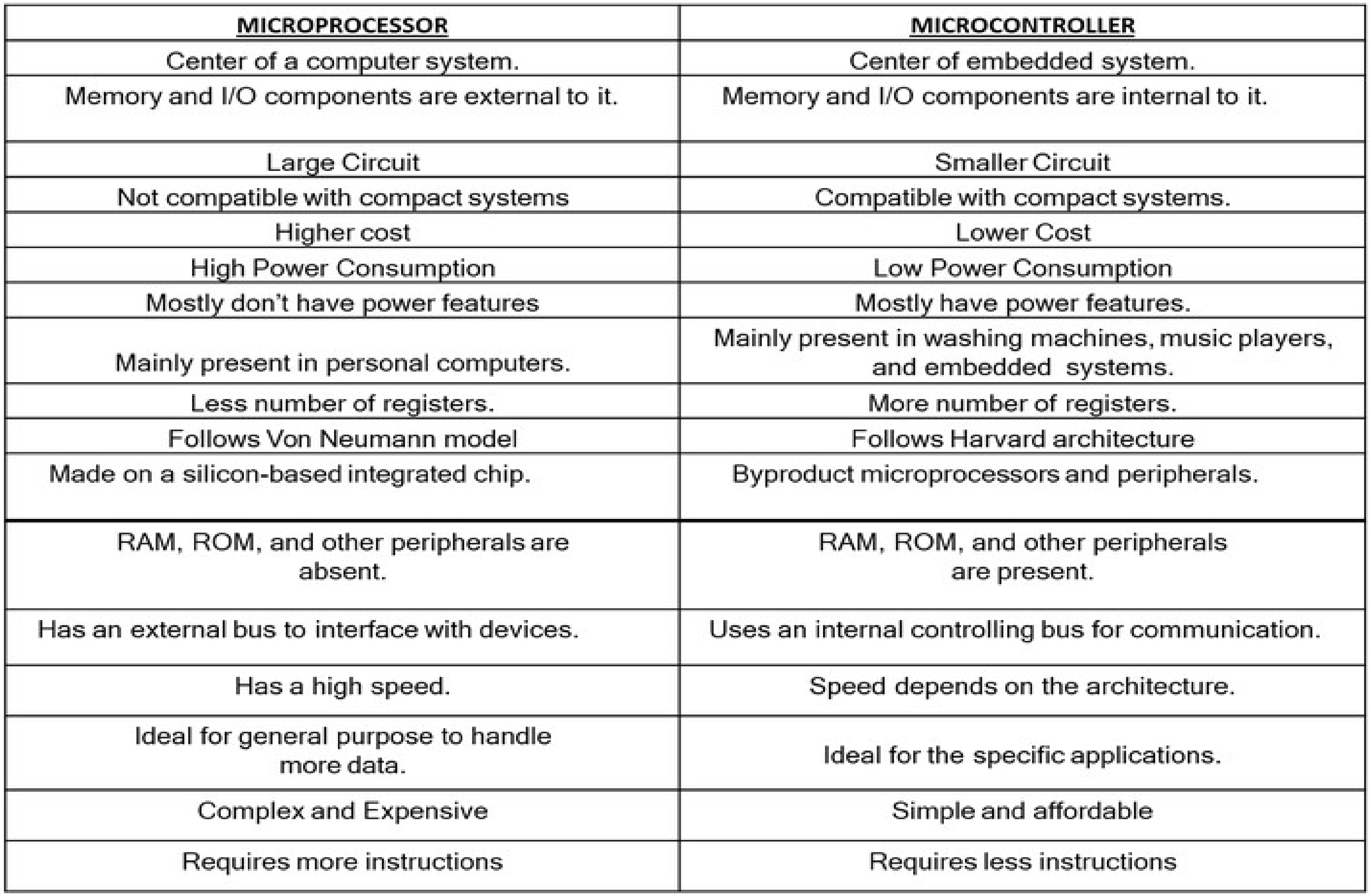


Figure 10. Timer/Counter 0 Mode 3: Two 8-Bit Counters

Interrupts

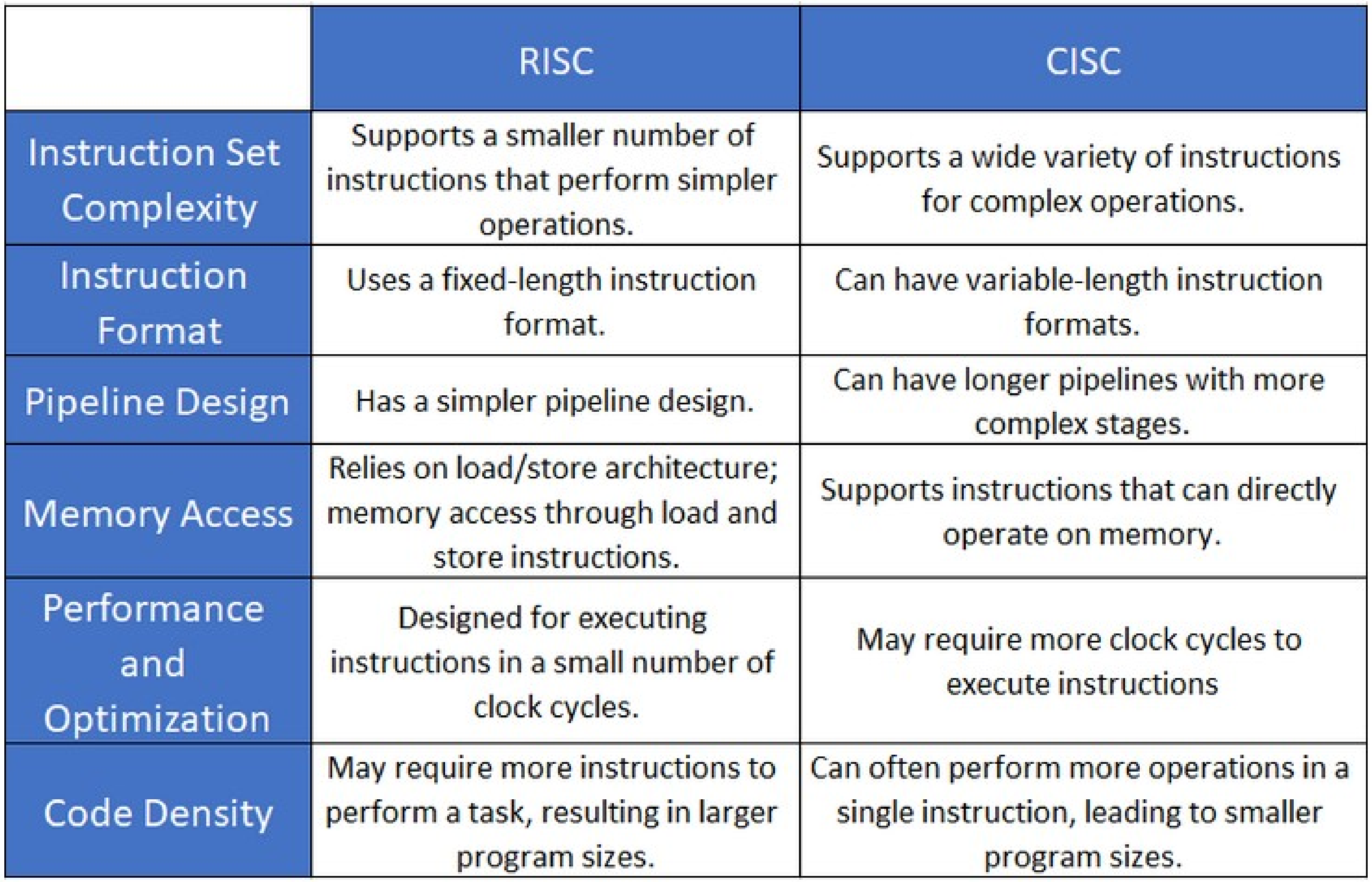
- Allow parallel tasking
  - Interrupt routine runs in “background”
- Allow fast, low-overhead interaction with environment
  - Don’t have to poll
  - Immediate reaction
- An automatic function call
  - Easy to program
- 8051 Interrupts
  - Serial port - wake up when data arrives/data has left
  - Timer 0 overflow
  - Timer 1 overflow
  - External interrupt 0
  - External interrupt 1

Microprocessor vs Microcontroller



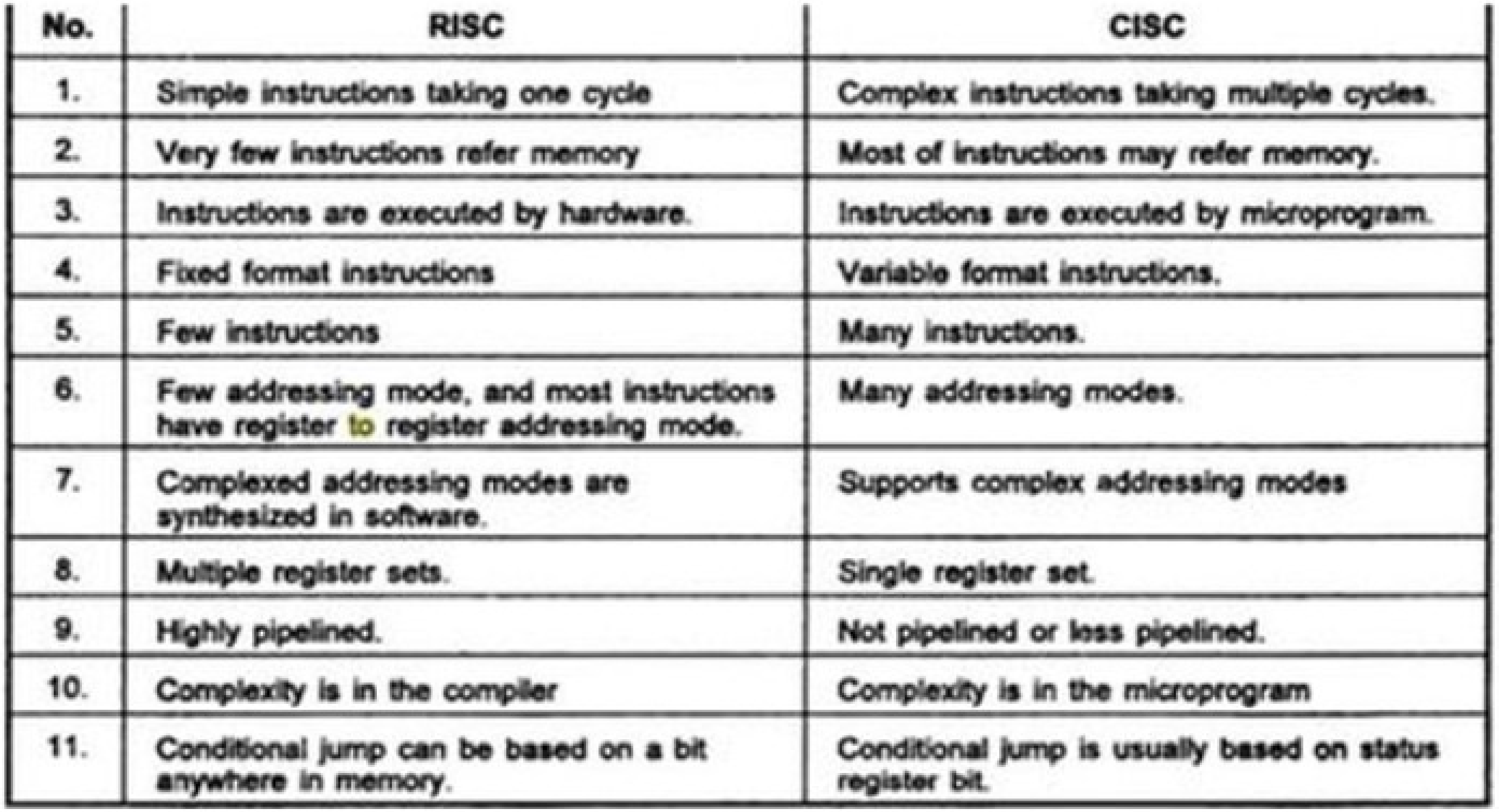
MICROPROCESSOR	**MICROCONTROLLER**
Center of a computer system.	Center of embedded system.
Memory and I/O components are external to it.	Memory and I/O components are internal to it.
Large Circuit	Smaller Circuit
Not compatible with compact systems	Compatible with compact systems.
Higher cost	Lower Cost
High Power Consumption	Low Power Consumption
Mostly don't have power features	Mostly have power features.
Mainly present in personal computers.	Mainly present in washing machines, music players, and embedded systems.
Less number of registers.	More number of registers.
Follows Von Neumann model	Follows Harvard architecture
Made on a silicon-based integrated chip.	Byproduct microprocessors and peripherals.
RAM, ROM, and other peripherals are absent.	RAM, ROM, and other peripherals are present.
Has an external bus to interface with devices.	Uses an internal controlling bus for communication.
Has a high speed.	Speed depends on the architecture.
Ideal for general purpose to handle more data.	Ideal for the specific applications.
Complex and Expensive	Simple and affordable
Requires more instructions	Requires less instructions

RISC vs. CISC … functional comparisons . . .



	RISC	CISC
Instruction Set Complexity	Supports a smaller number of instructions that perform simpler operations.	Supports a wide variety of instructions for complex operations.
Instruction Format	Uses a fixed-length instruction format.	Can have variable-length instruction formats.
Pipeline Design	Has a simpler pipeline design.	Can have longer pipelines with more complex stages.
Memory Access	Relies on load/store architecture; memory access through load and store instructions.	Supports instructions that can directly operate on memory.
Performance and Optimization	Designed for executing instructions in a small number of clock cycles.	May require more clock cycles to execute instructions
Code Density	May require more instructions to perform a task, resulting in larger program sizes.	Can often perform more operations in a single instruction, leading to smaller program sizes.

RISC vs. CISC refined comparisons



No.	RISC	CISC
1.	Simple instructions taking one cycle	Complex instructions taking multiple cycles.
2.	Very few instructions refer memory	Most of instructions may refer memory.
3.	Instructions are executed by hardware.	Instructions are executed by microprogram.
4.	Fixed format instructions	Variable format instructions.
5.	Few instructions	Many instructions.
6.	Few addressing mode, and most instructions have register to register addressing mode.	Many addressing modes.
7.	Complexed addressing modes are synthesized in software.	Supports complex addressing modes
8.	Multiple register sets.	Single register set.
9.	Highly pipelined.	Not pipelined or less pipelined.
10.	Complexity is in the compiler	Complexity is in the microprogram
11.	Conditional jump can be based on a bit anywhere in memory.	Conditional jump is usually based on status register bit.

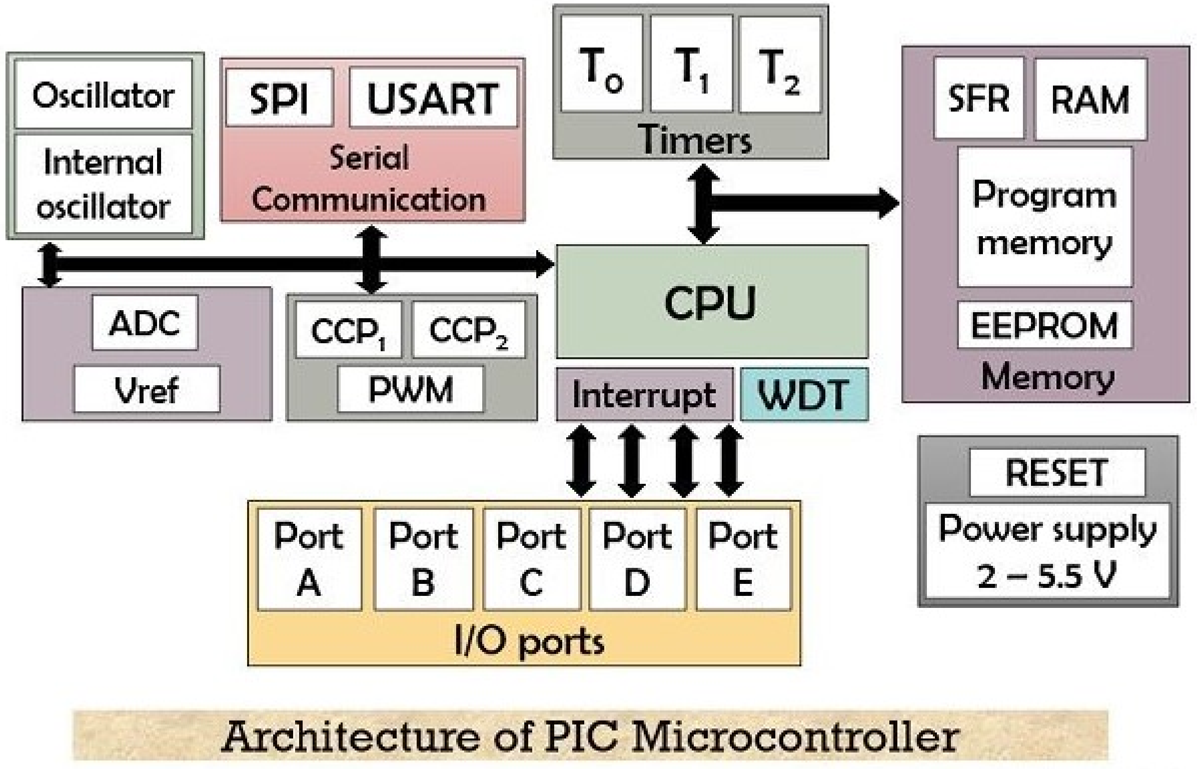
PIC Microcontroller

PIC is an abbreviation used for **P**eripheral **I**nterface **C**ontroller. PIC microcontroller is the smallest microcontroller in the world and are programmed to execute large number of operations. These were initially designed to support PDP (programmed data processor) computers, for controlling the peripheral devices. It is based on **RISC** architecture.

A microcontroller is nothing but a combination of processor, memory and peripherals in a single chip. In a similar way PIC microcontroller consists of data RAM with some hundred bytes of ROM for storing the desired program, some I/O ports, one timer on a single chip having 8 pins.

PIC Microcontroller block diagram

The figure below shows the architectural representation of PIC microcontroller:



PIC Microcontroller

CPU: CPU is the central processing unit that has ALU, CU, MU within it. The ALU performs arithmetic and logic operations according to the instructions received. The memory unit stores the instructions that are to be processed and also the data and instruction after being processed. While the control unit controls the I/O devices connected to the system.

As it is based on RISC architecture i.e., reduced instruction set computer, then it is to be noteworthy here that:

✓The number of instructions is small and approximately around a total of 35 instructions.

✓Due to less number of instructions, the operation performed by the CPU will be quite fast.

✓Processing of the instruction takes less time as the length of the instructions is small

✓RISC architecture also supports less complex compilation and easy debugging.

PIC Microcontroller

- **Memory**: Basically there exist two types of memory in PIC microcontroller-
  1. **Program or code memory:** As the name itself is suggesting that program memory holds the set of instructions that are desired to be performed by the microcontroller. It is basically referred as ROM. The memory space provided by this is 8K X 14 bits that can store 13-bit instruction or program. The PC accesses the program memory, and increments itself after fetching an instruction.

- **EEPROM:** ROM allows the program to be stored only once. But EEPROM i.e., electrically erasable programmable read only memory permits the multiple times writing of the code on the ROM.

- **Flash memory:** Flash memory is another PROM for this microcontroller. The program in the flash memory can be erased as many times as required.

PIC Microcontroller

- 2.Data memory: Data memory stores the data in it. Basically these are Random Access Memory i.e., RAM. It stores the data on a temporary basis in the registers. PIC microcontroller holds 368 bytes of RAM which is divided into banks. The register were the data is stored are classified as:

- General Purpose register: These are the registers that perform the general functions. Like addition, subtraction, multiplication etc. and further storing the results in other register. Hence no special function is assigned to these registers, the PIC holds the ability to directly access the data present in these registers.

- Special Purpose register: It is abbreviated as SFR. These registers are assigned some special functions by the processor and hence are not used for any general purpose. The operation to be performed by these registers are already set when the system is manufactured. Hence the functions of these registers are not variable.

PIC Microcontroller

Some important special function registers are:

- ⌘ STATUS register
- ⌘ TRIS register
- ⌘ PORT register

Like, the function of the **STATUS register** is to show the status of the program being performed.

I/O ports:

- ✓The number I/O ports is different for different PIC series.
- ✓The PIC16 series has 5 I/O port.
- ✓These are port A, port B, port C, port D and port E.

Bus:

- ✓Bus in this microcontroller is used to communicate between different units within it.
- ✓PIC has 2 types of buses, data bus and address bus.

The data bus transfers the data between memory and I/O unit. Whereas the address bus holds the address of the location to/ from where the data is loaded or fetched.

PIC Microcontroller

CCP module

- The name of CCP module represents capture / compare / PWM(pulse width modulation), which operates in three modes, such as capture mode, comparison mode and PWM mode.
- Capture mode: the capture mode captures the arrival time of the signal, or when the CCP pin changes to high level, it captures the value of Timer1.
- Comparison mode: the comparison mode is used as an analog comparator. When the Timer1 value reaches a reference value, it will generate an output.
- PWM mode: PWM mode provides pulse width modulation output with 10 bit resolution and programmable duty cycle.

PIC Microcontroller

Stack:

✓Since, the priority of critical interrupts are higher than the other tasks, whenever, an interrupt is generated then the processor of the PIC microcontroller must switch to handle the interrupt by stopping the operation which is being executed currently.

✓So, the stack stores the address of the program currently being in execution, until the PIC microcontroller handles the generated interrupt.

✓Once the interrupt handling is done, then the processor switches back to the main program whose address is stored in the stack.

Timers:

✓PIC microcontroller consists of 3 timers.

✓Out of the 3, 2 timers i.e., timer 0 and timer 2 are of 8-bit each while timer 1 is of 16 bit.

Advantages of PIC Microcontroller:

1. PIC microcontrollers are consistent and meantime failure percentage of PIC is very less.
2. The performance of the PIC microcontroller is very fast because of using RISC architecture.
3. When comparing to other microcontrollers, power consumption is very less and programming is also very easy.
4. Interfacing of an analog device is easy without any extra circuitry

Disadvantages of PIC Microcontroller:

1. The length of the program is high due to using RISC architecture (35 instructions)
2. One single accumulator is present and program memory is not accessible

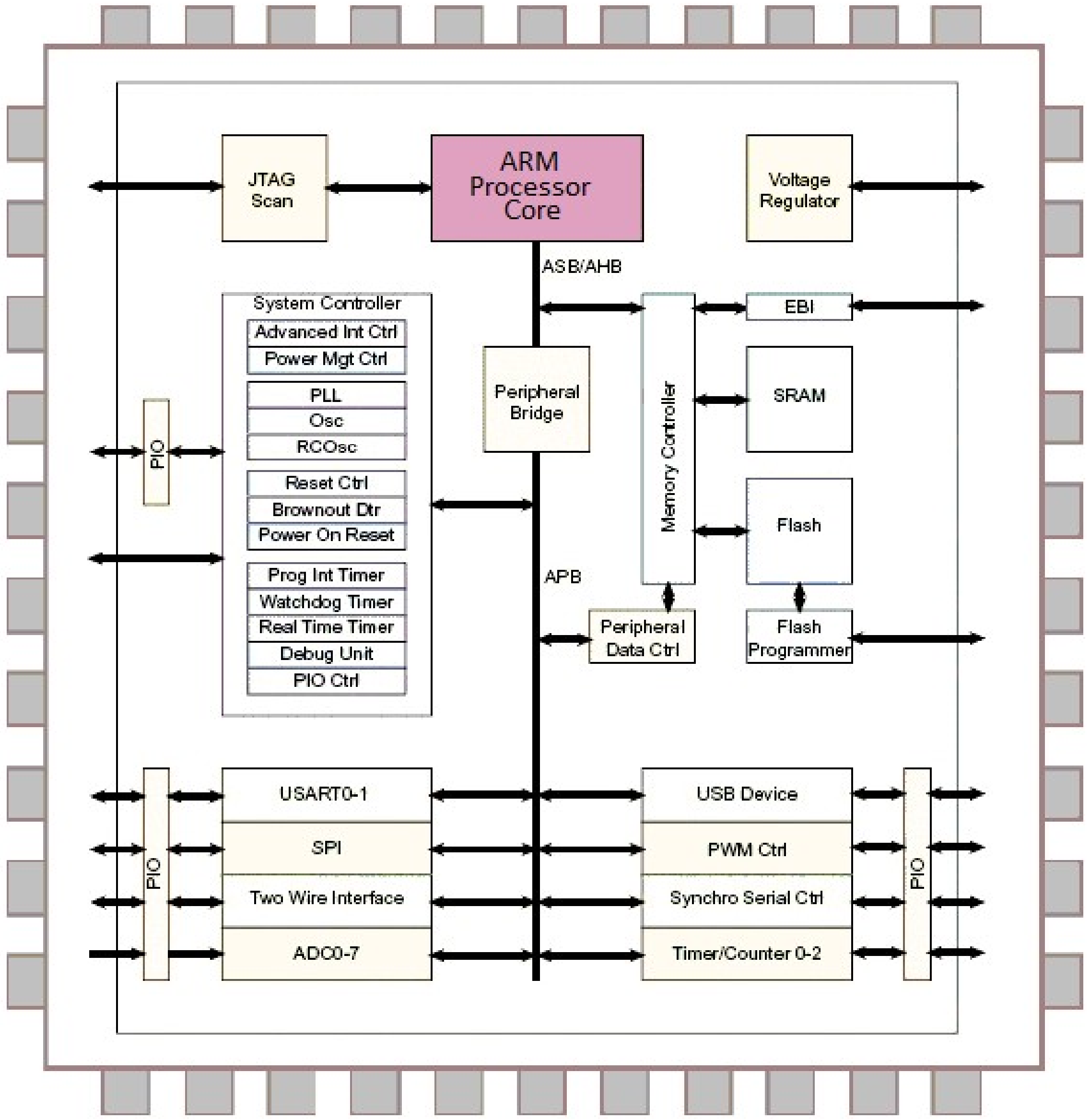
ARM Architecture

- ARM is an acronym for Advanced RISC Machines and is a family of RISC instruction set architectures (ISAs) for computer processors.
- It is the most pervasive processor architecture in the world, with more than 270 billion Arm-based chips have been shipped over the past three decades in products ranging from sensors, wearables and smart phones to supercomputers.
- The architecture has evolved over time, and version seven of the architecture, ARMv7, defines three architecture "profiles":
- A-profile, the "Application" profile, implemented by 32-bit cores in the Cortex-A series and by some non-ARM cores
- R-profile, the "Real-time" profile, implemented by cores in the Cortex-R series
- M-profile, the "Microcontroller" profile, implemented by most cores in the Cortex-M series

Benefits of the Arm CPU architecture include:

- Integrated security
- High performance and energy efficiency
- Large ecosystem for global support
- Pervasive across markets and locations
- Due to their low costs, low power consumption, and low heat generation, ARM processors are useful for light, portable, battery-powered devices, including smartphones, laptops, and tablet computers, as well as embedded systems

ARM Microcontroller



ARM Microcontroller

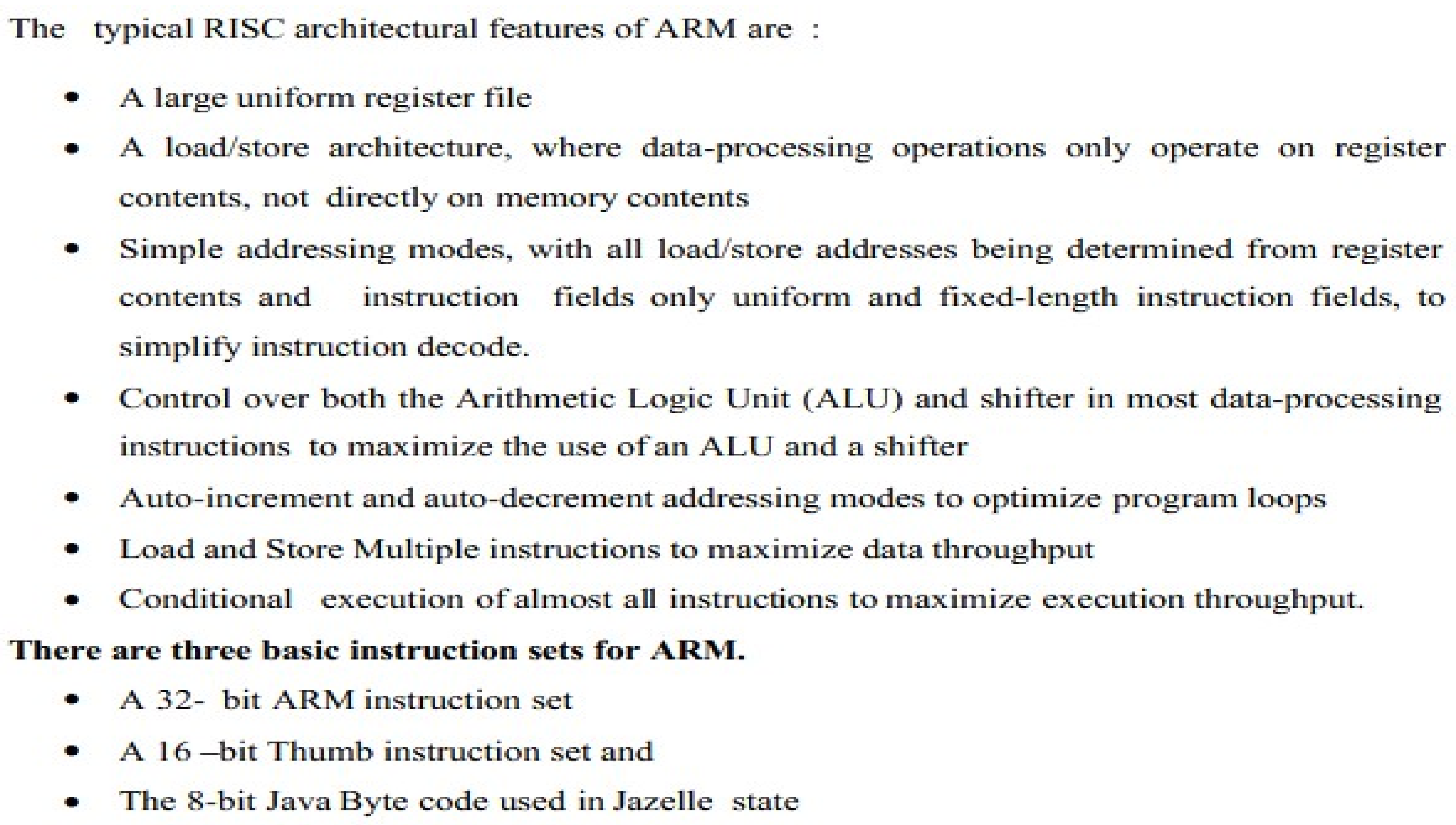
The ARM 7 processor is based on Von Neman model with a single bus for both data and instructions..(The ARM9 uses Harvard model).Though this will decrease the performance of ARM, it is overcome by the pipe line concept. ARM uses the Advanced Microcontroller Bus Architecture (AMBA) bus architecture. This AMBA include two system buses: the AMBA High-Speed Bus (AHB) or the Advanced System Bus (ASB), and the Advanced Peripheral Bus (APB).

The ARM processor consists of

- Arithmetic Logic Unit (32-bit)
- One Booth multiplier(32-bit)
- One Barrel shifter
- One Control unit
- Register file of 37 registers each of 32 bits.

In addition to this the ARM also consists of a **Program status register** of 32 bits, Some special registers like the **instruction register**, memory data read and write register and memory address register ,one **Priority encoder** which is used in the multiple load and store instruction to indicate which register in the register file to be loaded or stored and Multiplexers etc.

ARM Microcontroller



The typical RISC architectural features of ARM are :

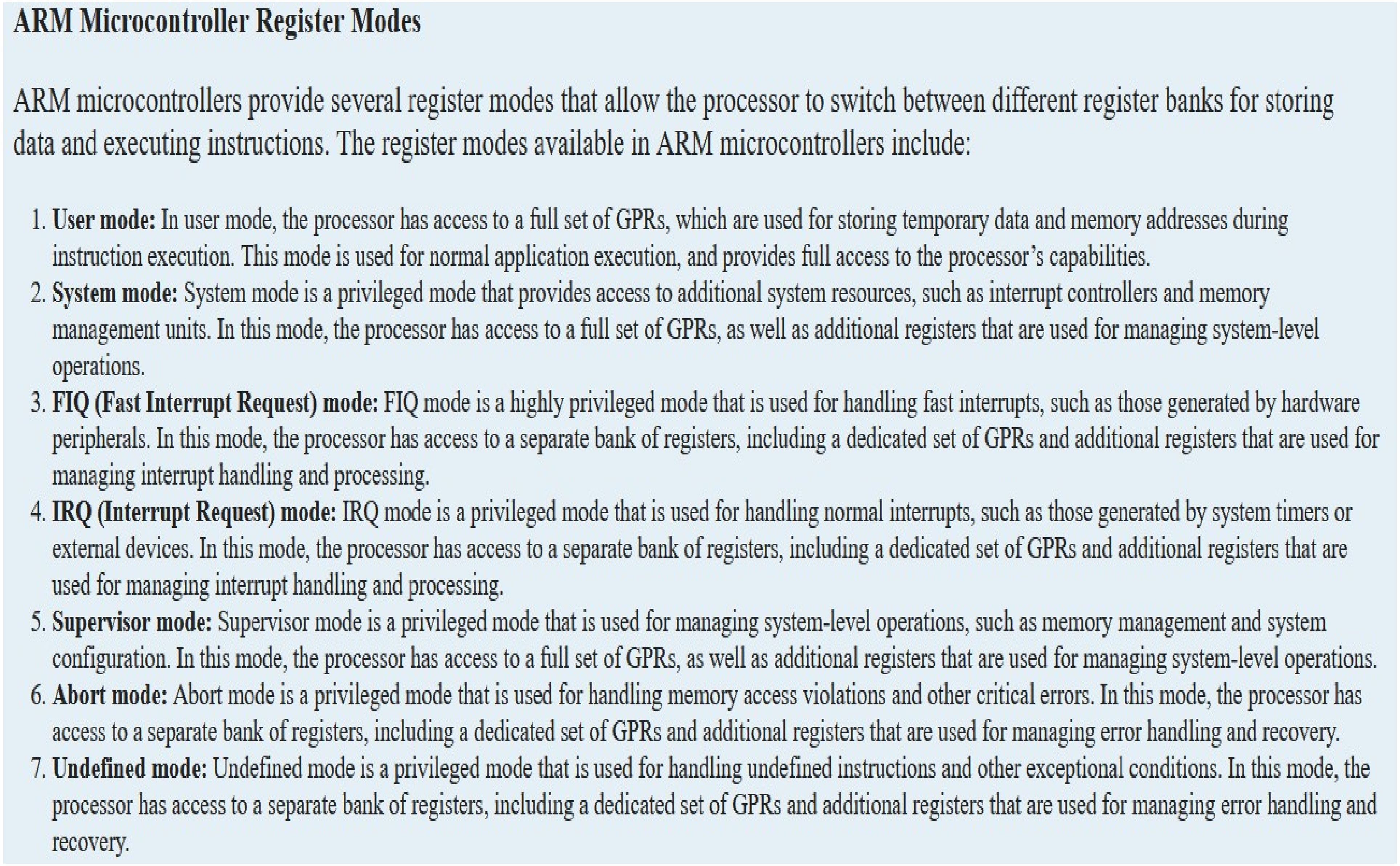
- A large uniform register file
- A load/store architecture, where data-processing operations only operate on register contents, not directly on memory contents
- Simple addressing modes, with all load/store addresses being determined from register contents and instruction fields only uniform and fixed-length instruction fields, to simplify instruction decode.
- Control over both the Arithmetic Logic Unit (ALU) and shifter in most data-processing instructions to maximize the use of an ALU and a shifter
- Auto-increment and auto-decrement addressing modes to optimize program loops
- Load and Store Multiple instructions to maximize data throughput
- Conditional execution of almost all instructions to maximize execution throughput.

There are three basic instruction sets for ARM.

- A 32- bit ARM instruction set
- A 16 –bit Thumb instruction set and
- The 8-bit Java Byte code used in Jazelle state

- **Jazelle DBX** (direct byte code execution) is an extension that allows some ARM processors to execute Java byte code in hardware as a third execution state alongside the existing ARM and Thumb modes

ARM Microcontroller



ARM Microcontroller Register Modes

ARM microcontrollers provide several register modes that allow the processor to switch between different register banks for storing data and executing instructions. The register modes available in ARM microcontrollers include:

1. **User mode:** In user mode, the processor has access to a full set of GPRs, which are used for storing temporary data and memory addresses during instruction execution. This mode is used for normal application execution, and provides full access to the processor's capabilities.
2. **System mode:** System mode is a privileged mode that provides access to additional system resources, such as interrupt controllers and memory management units. In this mode, the processor has access to a full set of GPRs, as well as additional registers that are used for managing system-level operations.
3. **FIQ (Fast Interrupt Request) mode:** FIQ mode is a highly privileged mode that is used for handling fast interrupts, such as those generated by hardware peripherals. In this mode, the processor has access to a separate bank of registers, including a dedicated set of GPRs and additional registers that are used for managing interrupt handling and processing.
4. **IRQ (Interrupt Request) mode:** IRQ mode is a privileged mode that is used for handling normal interrupts, such as those generated by system timers or external devices. In this mode, the processor has access to a separate bank of registers, including a dedicated set of GPRs and additional registers that are used for managing interrupt handling and processing.
5. **Supervisor mode:** Supervisor mode is a privileged mode that is used for managing system-level operations, such as memory management and system configuration. In this mode, the processor has access to a full set of GPRs, as well as additional registers that are used for managing system-level operations.
6. **Abort mode:** Abort mode is a privileged mode that is used for handling memory access violations and other critical errors. In this mode, the processor has access to a separate bank of registers, including a dedicated set of GPRs and additional registers that are used for managing error handling and recovery.
7. **Undefined mode:** Undefined mode is a privileged mode that is used for handling undefined instructions and other exceptional conditions. In this mode, the processor has access to a separate bank of registers, including a dedicated set of GPRs and additional registers that are used for managing error handling and recovery.