Name: Salma Sameh Hassan Mohamed

Task_7_ES

Embedded Systems Concepts

000_video: Contents

001_video:

Computing System : consist of three main methods
1) processor
2) memory
3) I /O

consist of two types
1) General purpose
2) specific purpose

Parts that should be paid attention to during EM sys challenge
Size, power consumption,cost,speed or time.

There are two ways to manufacture ES
1) System board(SB): add system on the board.
2) System on chip (Soc): IC coated with epoxy.

Compare between SB & Soc

	SB	Soc
Size	Large	Small
Cost	High	Low
Power consumption	High	Low
Performance	Equal	Equal

IC: integrated circuit ,It is a single chip that has a specific function.

VLSI: very large scale IC.
Moore's Law

MPU: micro processor unit : is three named
1) processor
2) micro processor
3)CPU

Compare between MPU & MCU

MCU: inside(MPU, memory,I/O,GPU,DSP)
MPU: is apart of MCU

Secondary Processor

GPU: it understands the instructions related to graphics.
DSP: use It uses complex mathematical operations such as Fourier series

To write code use to application

1) **Bare Metal software**
2) **os Application**

ECU:is considered SOC adding sensors and actuators.

002_video:

computing System

1)CPU
2) memory
3)I/O

1) **Processor: consist of three main methods**
1)ALU
2) **CU**
3) **Register files**

	RISC	CISC
Instruction decoder	Few	much
Software	need strong software	need simple software
Cost	High sw, low Hw Same	low Hw, High sw same
Performance	Hw high, sw low same	Hw low, sw high same
Power consumption	ALU low,ID high	ALU high ,ID low

Compare between General purpose Reg & Special purpose Register.

Special Purpose Register:
1) **PC: toward the next instruction.**

2) SP: toward the last position in stack or
the next position.

3) ACC: in the old MP, it stored the result but Temporarily.

4) IR: store the instruction coming from memory after fetch.

4)PSW: it is A collection of flags.

2) Memory: is considered of collection of some locations, the size of locations depends on the Architecture.

Note:access time(write) > access time(read)

2) D-flip-flop:is less element to keep zero or ones.

3) organization: it enable to make 16K using
Four of 4K.

003_video: memory

Types of memory
1) volatile:RAM
2) non volatile: ROM
3) Hybrid: it is mix between RAM and ROM

1) volatile memory:RAM
*Read and write memory is faster than ROM,RAM is called RWM (working memory).
Types of RAM:

	SRAM	DRAM
Size	Low	High
Cost	High	Low
Performance	High	Low
Power consumption	Low	High

Note:
*DRAM consists of a capacitor and simple MOSFET.

*SRAM based on transistor,used at least 6 transistors (D-flip-flop).

004_video: memory

2) nonvolatile memory:ROM
*Read only memory or program memory.

Note: access time for ROM >access time for RAM so ROM is slower than RAM.

In manufacturing ROM depends on floating gates MOSFET (FGM).

Erasing state: act as no charge so the bit is one.
Programming state: inside the FGM the charge is negative so the bit is zero.

Types of ROM
1)Mask programming ROM
2)PROM
3) EPROM

1) MASK:
called OTP (only time program)Such as toys,Bios Chip(inside the maze board),The factory burns the code on it.

2) PROM:
Programmable ROM, called OTP but The user burns the code on it, consist of fuses.If you burn the fuse, it will be considered zero otherwise 1.

3) EPROM:
Erasable programmable ROM, Erasable for ultra violet.
Advantages:
1) Erasable
2) save data for a long time.
disadvantages:
Noise and Radiation affect the data inside it, so corruption will occur.

---

3) Hybrid: take the best from RAM(can be Read and Write) & the best from ROM(Non-volatile).

Types of ROM:
1) E^2 PROM
2) flash
3)NVRAM

1) E^2 PROM:
Electrical Erasable Programming ROM, high Endurance 100,000 times, byte access,high cost per bit so low density, example store password.

2) Flash:

It's faster than E^2 PROM because it blocks access sector by sector so it's cheaper than E2PROM, Endurance 10,000 times so low cost per bit,high density.

***The problem of flash is that you should write in The whole sector.** **But Autozar solved that by using the layer to emulate the flash.**

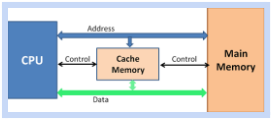
3) NVRAM:

non volatile RAM
***SRAM+battery**
***SRAM+E^2 PROM+small battery (Transfers important parts from SRAM to E2PROM when the battery is about to run out**
High cost per bit

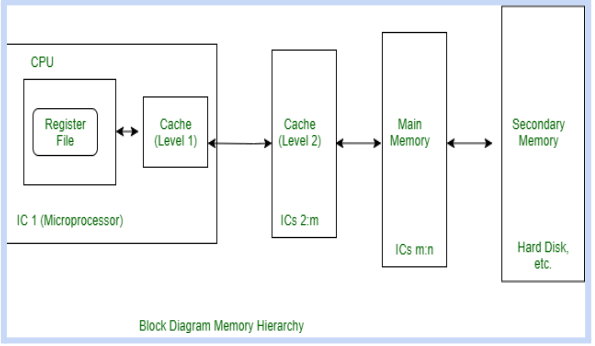
Task_8_ES

005_video: cache memory

Cache memory: According to the embedded system cache, it is called SRAM, high cost per bit, It is located between the CPU and main memory,maybe has more levels.



***For example, this level stays inside the MPU and another level stays outside the MPU.**



How does cache work?

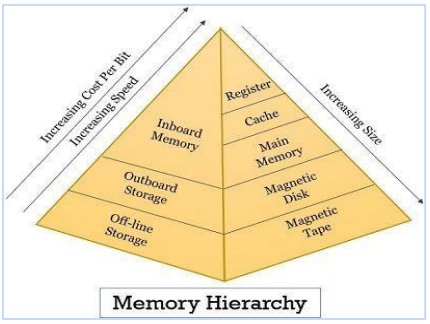
When the CPU requests a specific address, the cache will return a range of addresses because the CPU needs them after that.

cache hit : **This means the address inside the cache.**

cache miss: **The address was not found in the cache.**

Note:
cache=cache memory+controller.

Register files: **It is a memory that is considered the fastest thing in all types of memories,why?Because it does not need to put an address on the address bus.**

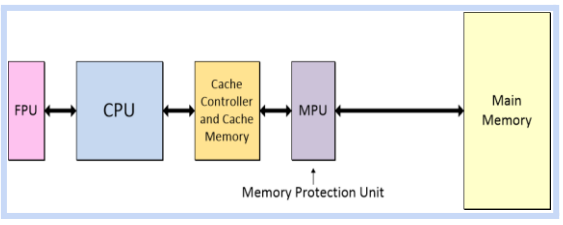


As for speed:
CPU >>Register files>>cache in core >> cache inside MPU >> cache outside MPU>>RAM>>Hard disk.

cache coherence: **If something changes in one cache, its controller will tell the rest of the controllers of the other cache.**

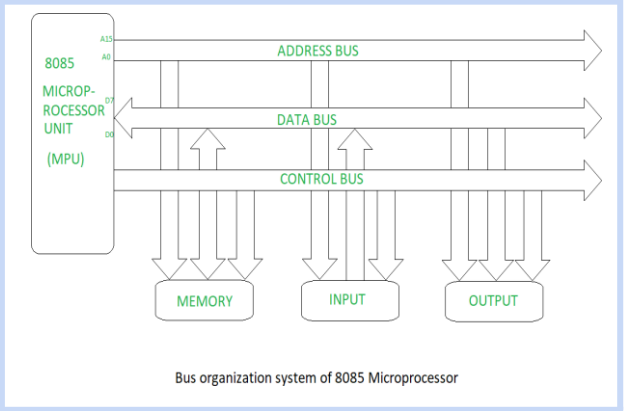
FPU: **Floating point unit, inside or outside the MPU,Available inside or outside the MPU has the advantage that it improves the performance of the MPU,such as interrupt controller.**

MPU: **memory protection unit,**



3) I/O : **This is the peripheral that is connected to the CPU, for example GPIO: general purpose input output(It controls the input pins or the output).**

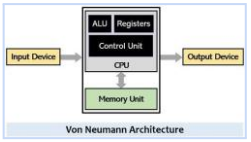
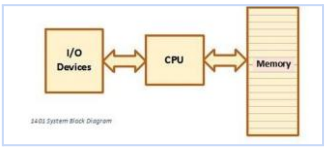
Bus set		
Data bus	Address bus	Control bus
Pio direction	Uni direction	It consists of a number of lines



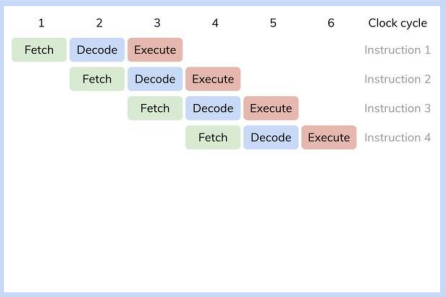
Bus organization system of 8085 Microprocessor

006_video: Arch

Lec:2

Architecture		
	Von neumann	Harvard
Called	one memory system	Port Mapped
Form	 Von Neumann Architecture	
used in	PC	in microcontroller
Pipeline	Can't Support	Support

Pipeline: Is that I can fetch, decode, and execute in one cycle? If you follow the pipeline, then parallelize the instructions, so the execution time will decrease and the performance of the system will increase.



Note:

1)RISC:can support pipeline,Because most of the instructions in RISC are executed in one cycle cycle.

2)CISC: cannot support pipelining because instructions in CISC terminate in more than one cycle.

Clock System		
	Electrical RC oscillator	Mechanical
Definition		It is a material that conducts electricity that produces a square wave

Type in clock system	Electrical	Mechanical	
	RC-oscillator	Ceramic Resonator	Crystal oscillator
cost	Low,Because it is an oscillator consisting of a capacitor, resistor and a 555 timer ✓	in between	High because consists of crystal
Accuracy	Low x	in between	High ✓
Settling Time	High x	in between	Low ✓

Note:

choosing the type of system depends on noise immunity,If you want to put the system in a car coach, you should use RC oscillator,Because ceramic and crystal do not mind noise and vibration.

Noise immunity

1) Temperature
2)EMI(electromagnetic interface)
3) vibration

Noise immunity			
	RC oscillator	Ceramic	Crystal
Temperature immunity	Low x ,Because of the resistance in the circuit.	immunity ✓	High immunity ✓
EMI immunity	Low x ,The resistor and capacitor are affected by electromagnetic waves	immunity ✓	High immunity ✓
Vibration immunity	High ✓	Low x	Low x, Because it works with vibration.

007_video: connection methods

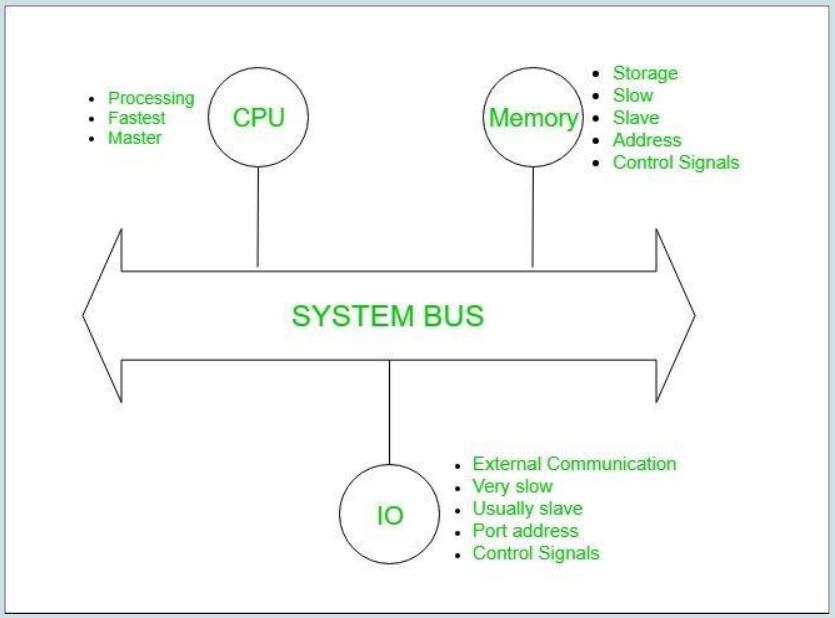
1) **Memory mapped:** used in Embedded System,each device has a range of addresses**(base address + offset)**.

base address: This connects me to the peripheral such as **GPIO**.

offset : To connect me to all the registers inside the peripheral.each register has a different offset.

Register Types

1) output data register
2) input data register
3) direction register



digital design

System bus : There are types, for example AXI (standard protocol),it has the master in the CPU and the peripheral I/O out of it.it enters the master from the CPU and exits the selev to the peripheral(GPIO).

Peripheral:(Hardware circuit + Register)

Hardware circuit: It performs the operation of the GPIO.

Register: You deal with it to turn something on, turn it off, or write 0 or 1 in it.

TRM: technical reference manual,you should read it.

Memory mapping:It contains all the peripheral and its base addresses,It identifies the start and end of each peripheral.

Table 4-16. Boot ROM Memory Map (Silicon revision 0,A)

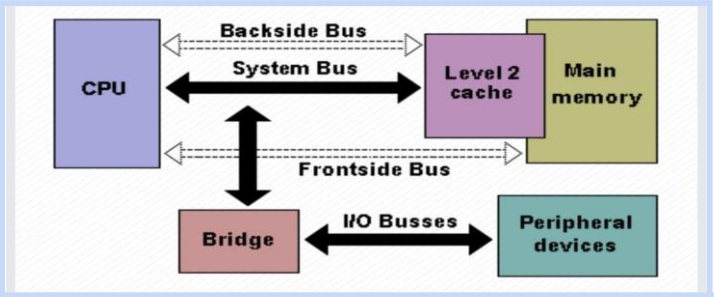
Memory	Start Address	End Address	Length
ROM Signature	0x003F 0000	0x003F 0001	0x0002
TI-RTOS (ROM)	0x003F 7200	0x003F 91FF	0x2000
FPU32 Table	0x003F 9200	0x003F A9FF	0x1800
Boot	0x003F AA00	0x003F E9CF	0x3FD0
Unsecure ROM Checksum	0x003F E9D0	0x003F EA11	0x0042
CPU Spintac Data(1)	0x003F EA12	0x003F EA21	0x0010
CPU Fast Data(1)	0x003F EA22	0x003F EB21	0x1000
PIE Mismatch Handler	0x003F FF22	0x003F FF71	0x0050
CRC Table	0x003F FF72	0x003F FF79	0x0008
Version	0x003F FF7A	0x003F FF7B	0x0002
Vectors	0x003F FFBE	0x003F FFFF	0x0042
TI-RTOS (Flash)	0x0008 1010	0x0008 13EE	0x3FDF

(1) Check the data manual to determine if these are available for your device part number. If not available, treat these sections as reserved.

Table 4-17. Boot ROM Memory Map (Silicon revision B)

memory mapping

Bus bridge: It converts the system bus from one type to another, so that the master and the selev must remain of the same system bus type.



008_video:

Example: write 1. in bit 16 in Register ODR

bit 16

[redacted]	1	[redacted]

GPIO_ODR |=(1<<16)
Or
using numeric memory address directly

```
#define GPIO_ODR (base_address + offset)

void main() {
   *((volatile uint32_t*) GPIO_ODR) |= (1 << 16);
}
```

using d_reference in #define

```
#define GPIO_ODR (*(volatile uint32_t *)(0x00008 + base_address))
GPIO_ODR |= (1 << 16);
```

using structure & pointer for all register

```
#define GPIO ((volatile unsigned int*)0xADDRESS_OF_GPIO_REGISTER)
*GPIO |= (1 << 16);
```

using structure & union & pointer

```
#include <stdint.h>

typedef struct {
   volatile uint32_t ODR;
} GPIO_TypeDef;

#define GPIOA ((GPIO_TypeDef *)0x50030)
```

```
int main() {
   GPIOA->ODR |= (1 << 16);
   return 0;
}
```

Reusability: ST took the ARM MPU. They made a board called stm32Flo3c6 on it.

ST:

1) TRM: general manual For all the family
2) data sheet : Especially for family flo3.

If you want the speed of the CPU in stm32Flo, I will open the block diagram for stm32Flo3xx and look at the CPU and you will find the max 72MHZ.

APB: advanced peripheral bus

Note:

*The system bus is the same speed as the CPU.

*The higher the speed, the higher the power consumption, which is why two buses with different speeds are used: APB1 and ABP2.

*In some peripherals you don't need high speed, the highest is 36MHz which will be connected to APB2 bus.

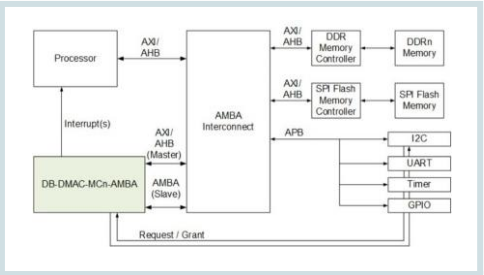
*frequency of APB1(72MHz)>>frequency of APB2(36MHZ).

Hardware port Types

1) two types:

A) master: transshiated transaction

B) Slave: Sends the data for size.



DMA: Direct memory Access,has master on the bus, it helps the CPU to perform a specific function without me occupying the CPU.

AMBA: **The ARM Advanced Microcontroller Bus Arch ,It is standard. Its function is to make a connection between the peripheral inside the chip, connecting to different types of buses.**

Note :

It is important that you know the buses' bandwidth and laterancy.

bandwidth: I can send a bit per second on the bus(It is present in the block diagram).

laterancy: The time the process takes during implementation.The less laterancy, the better the bus.