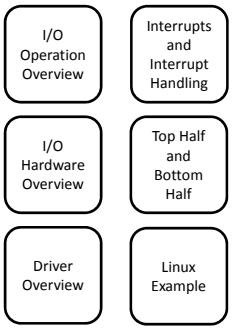
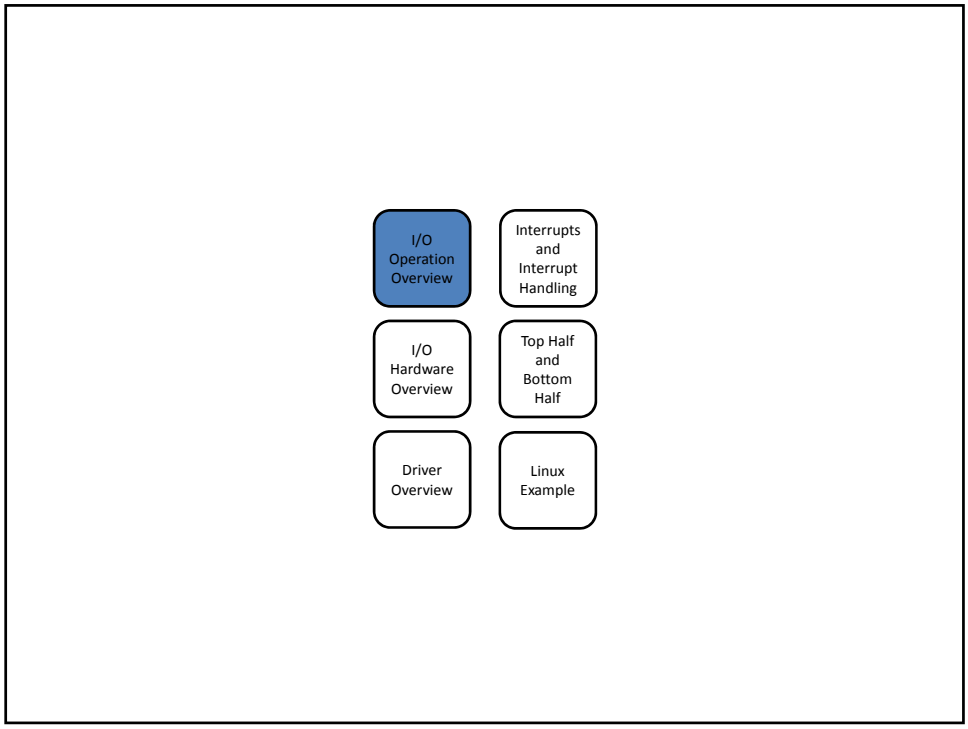
Operating System Design and Implementation

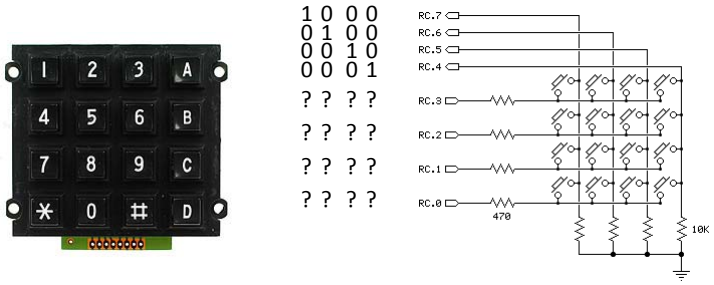
Interrupt and Interrupt Handling

Shiao-Li Tsao

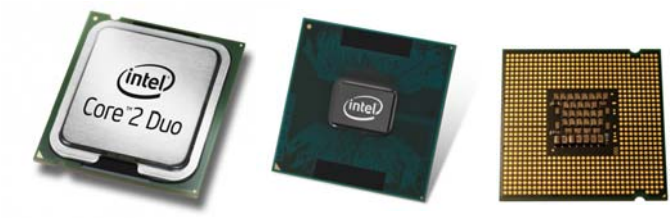




Step 1: A keypad



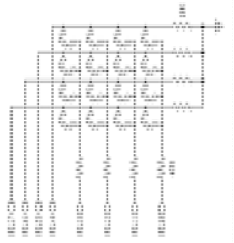
Step 2: Connect your keypad to your X86



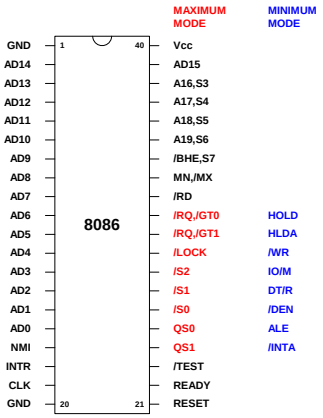
>450 pins

8086: 40 pins

Step 2: Connect your keypad to your X86 (Cont.)

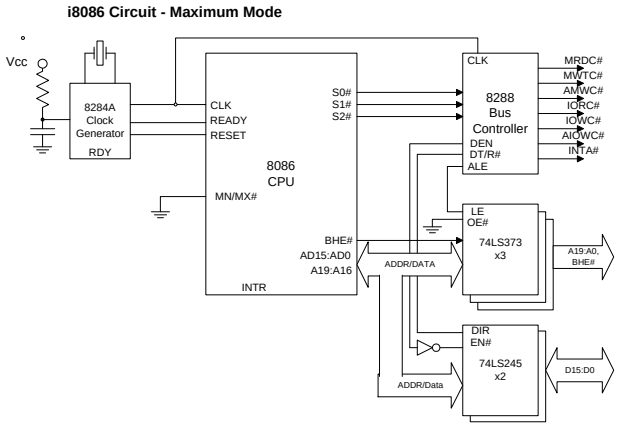


MOV [0x1234], 0001b
MOV ax, [0x5678]

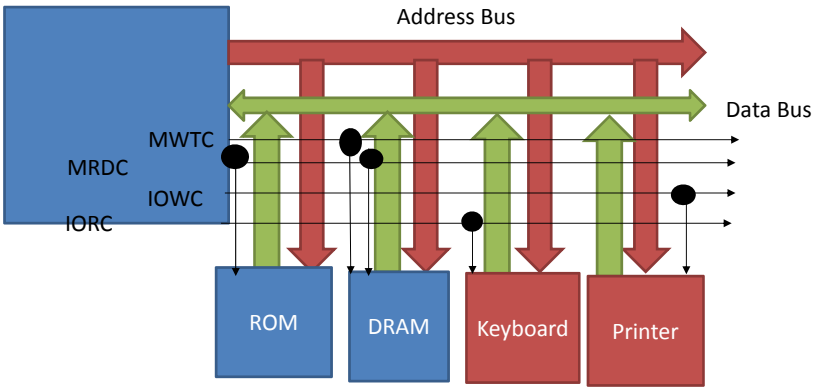


Step 2: Connect your keypad to your X86 (Cont.)

- 8086 and related chipsets

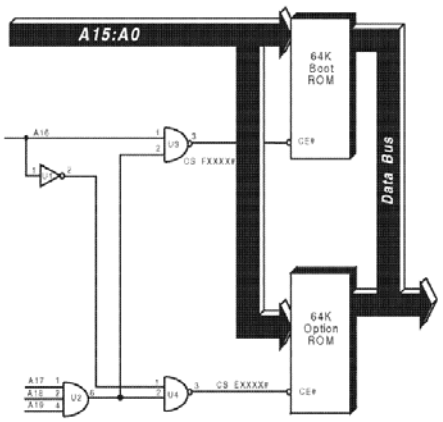


Step 2: Connect your keypad to your X86 (Cont.)



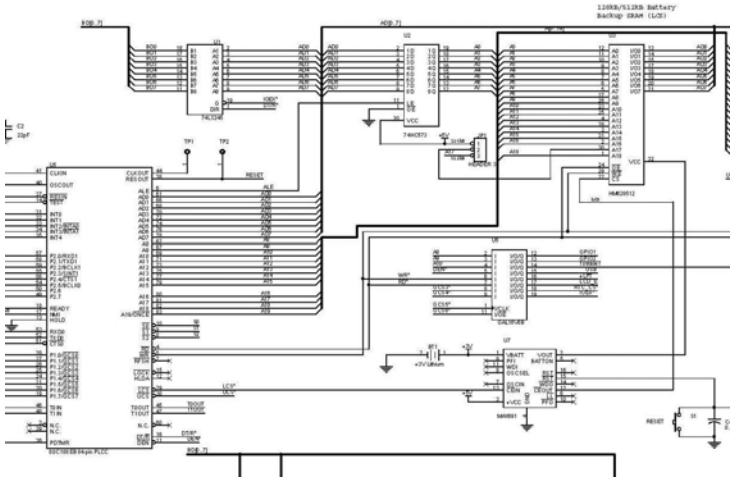
Ref: http://www.csee.umbc.edu/~cpatel2/links/310/

Step 2: Connect your keypad to your X86 (Cont.)

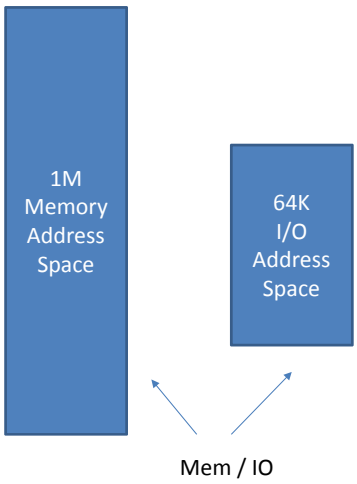


Source: ISA System Architecture 3rd edition

Step 2: Connect your keypad to your X86 (Cont.)



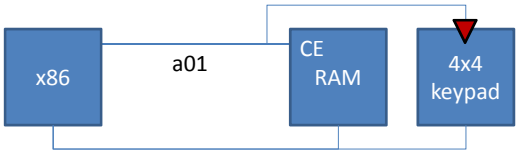
Step 2: Connect your keypad to your X86 (Cont.)



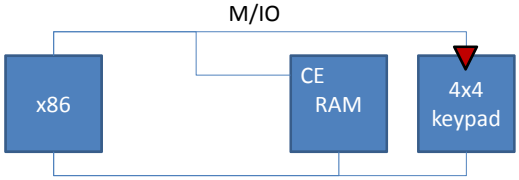
Ref: ISA System Architecture 3rd edition

Step 2: Connect your keypad to your X86 (Cont.)

- Mapping to an address



- Mapping to a I/O port



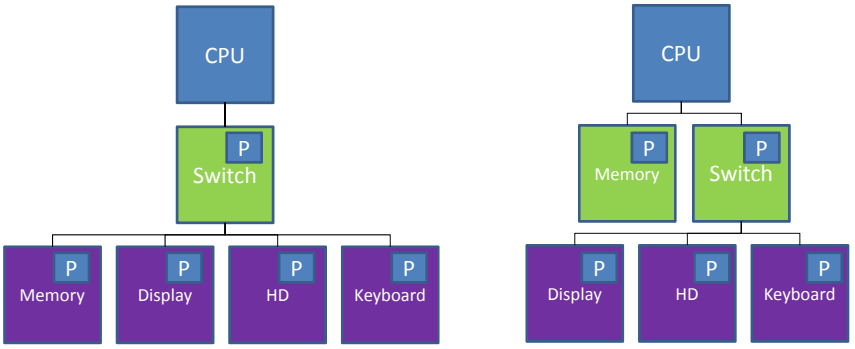
Step 3: Write an x86 program/read the inputs

- Assembly codes
  - Mapping to an address
    - MOV
  - Mapping to a I/O port
    - IN/OUT

How about?

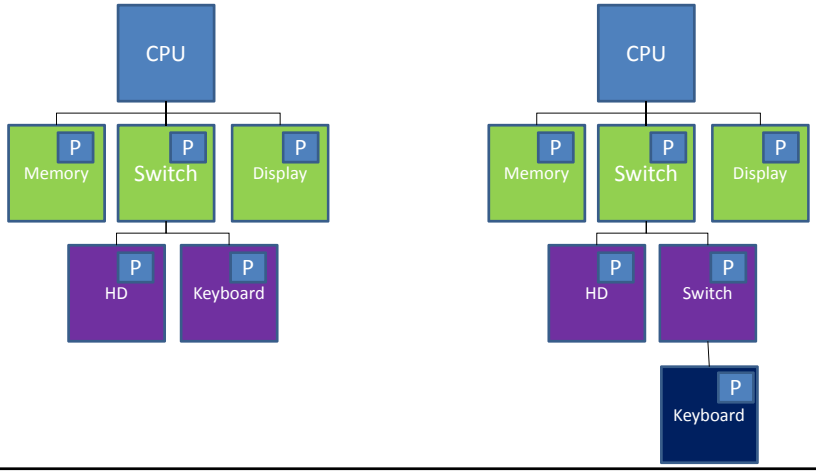
14

How can we solve the problems?



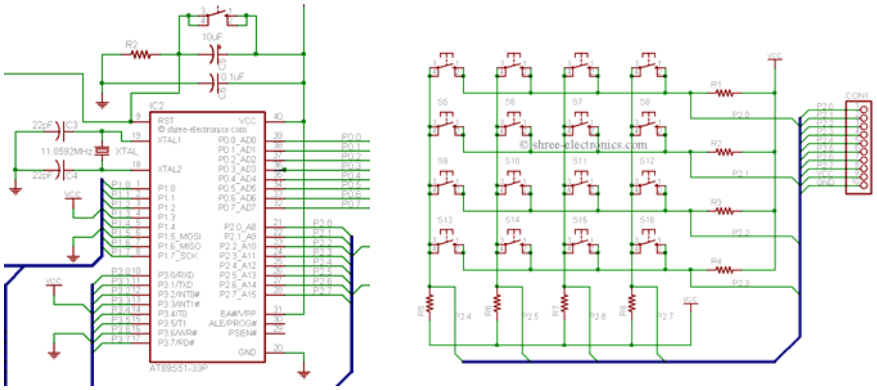
15

How can we solve the problems?



16

Step 4: Connect your keypad to 8051



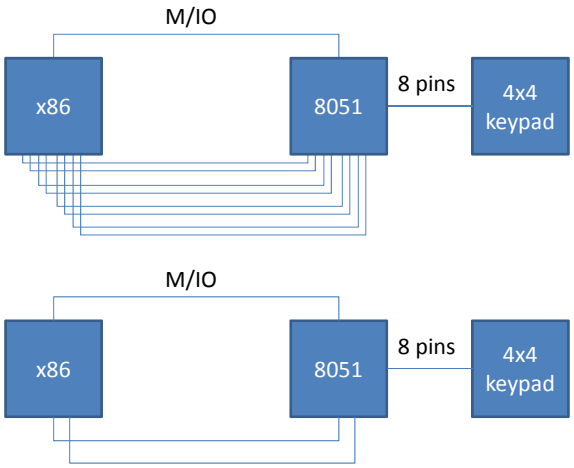
17

Step 5: Write an 8051 program

```
scan_key:mov P2,#0ffh
      clr P2.4
      mov a, P2
      anl a,#00001111b
      cjne a,#00001111b,Row0
      setb P2.4
      clr P2.5
      mov a, P2
      anl a,#00001111b
      cjne a,#00001111b,Row1
      setb P2.5
      clr P2.6
      mov a, P2
      anl a,#00001111b
      cjne a,#00001111b,Row2
      setb P2.6
      clr P2.7
      mov a, P2
      anl a,#00001111b
      cjne a,#00001111b,Row3
      setb P2.7
      ret
```

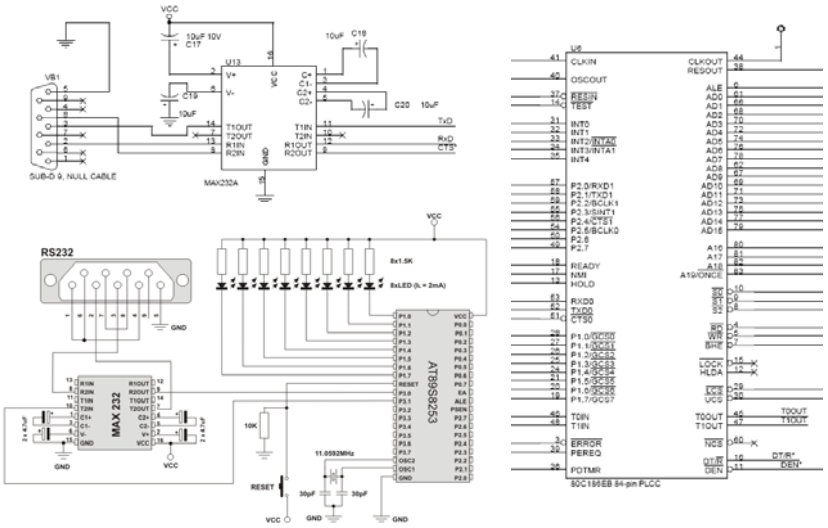
18

Step 6: Connect your keypad+8051 to your PC



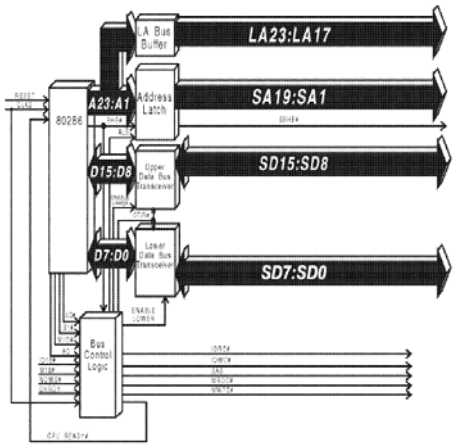
19

Step 6: Connect your keypad+8051 to your PC (Cont.)



20

Step 6: Connect your keypad+8051 to your PC (Cont.)



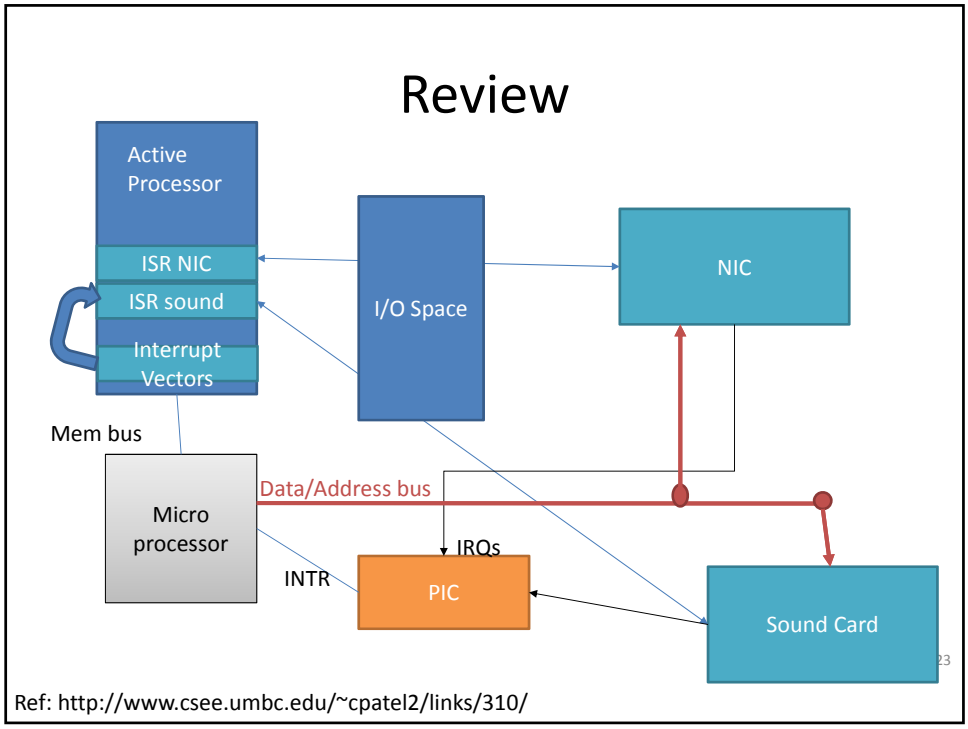
21

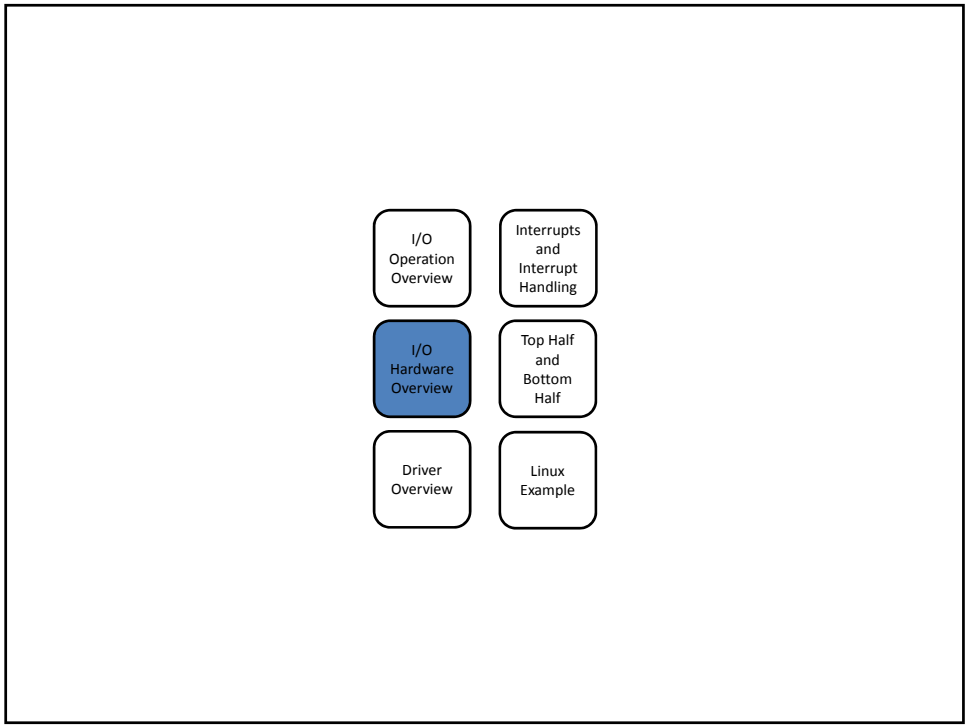
Step 6: Connect your keypad+8051 to your PC (Cont.)

ISA UART card

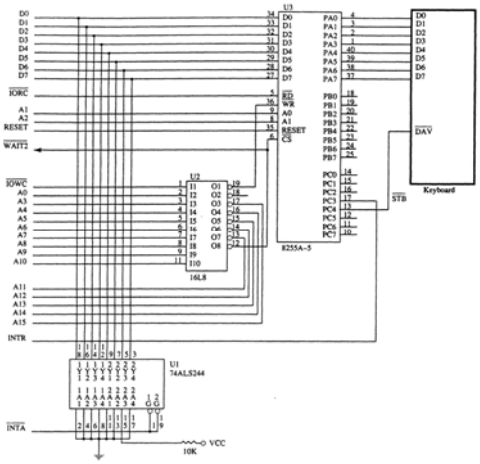
PCI UART card

22


Review
Active Processor
ISR NIC
ISR sound
Interrupt Vectors
I/O Space
NIC
Mem bus
Micro processor
Data/Address bus
IRQs
INTR
PIC
Sound Card
23
Ref: http://www.csee.umbc.edu/~cpatel2/links/310/

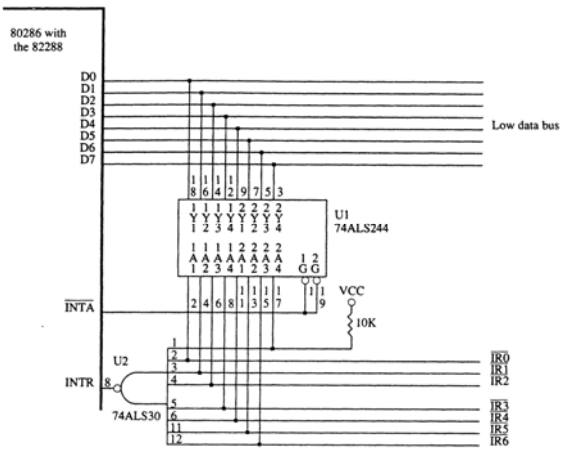

I/O Operation Overview
Interrupts and Interrupt Handling
I/O Hardware Overview
Top Half and Bottom Half
Driver Overview
Linux Example

Interrupt and keyboard



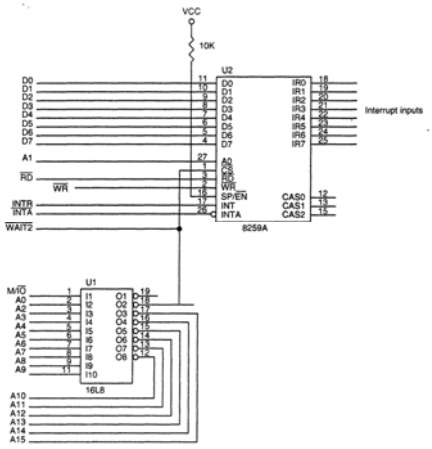
25

Expanding interrupts



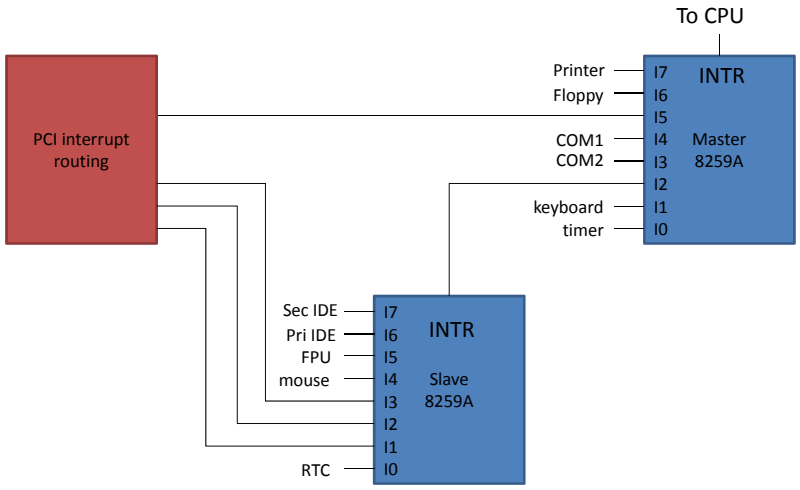
26

8259 programmable interrupt controller

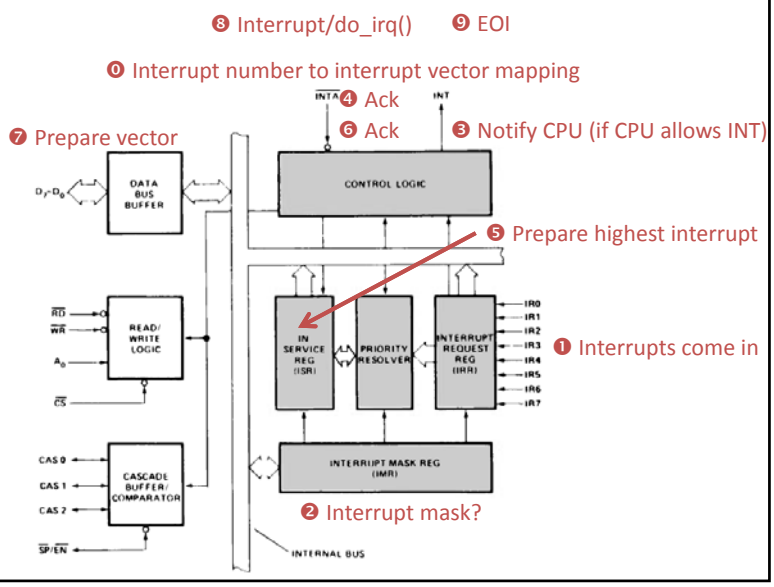


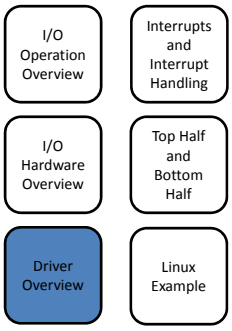
27

PIC

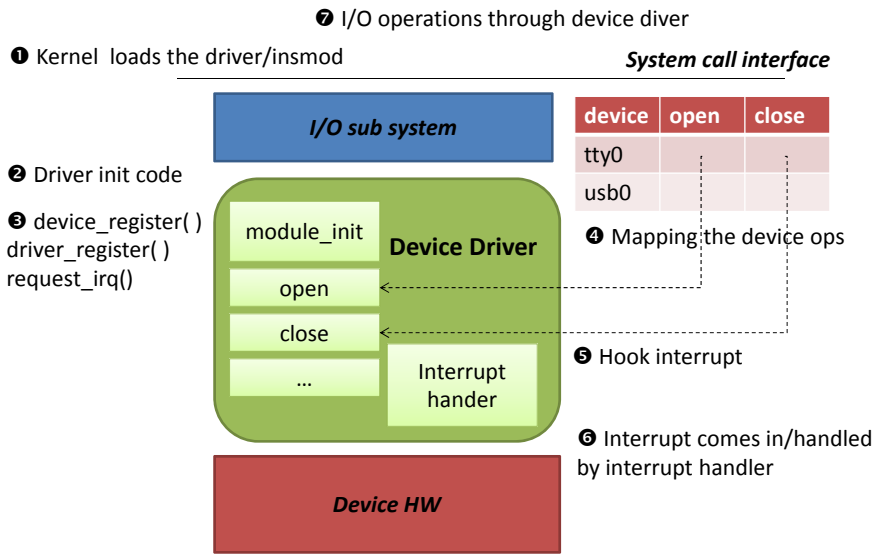


PIC Interrupt Procedure

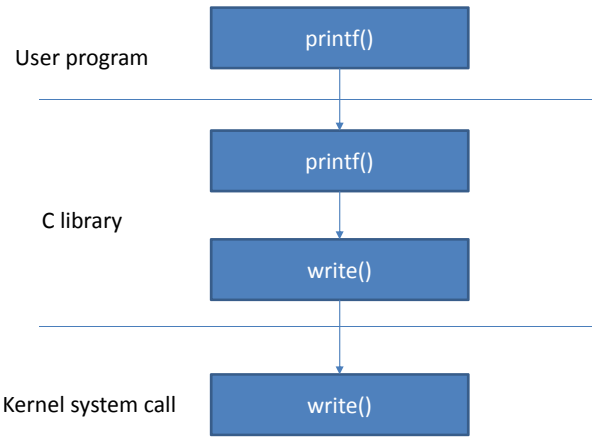




Driver Basics

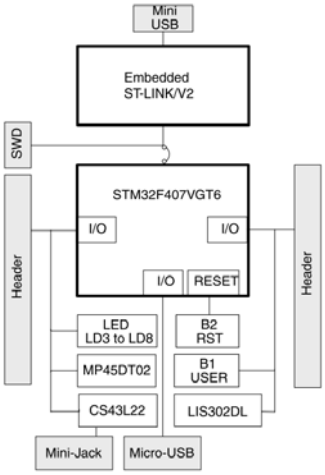


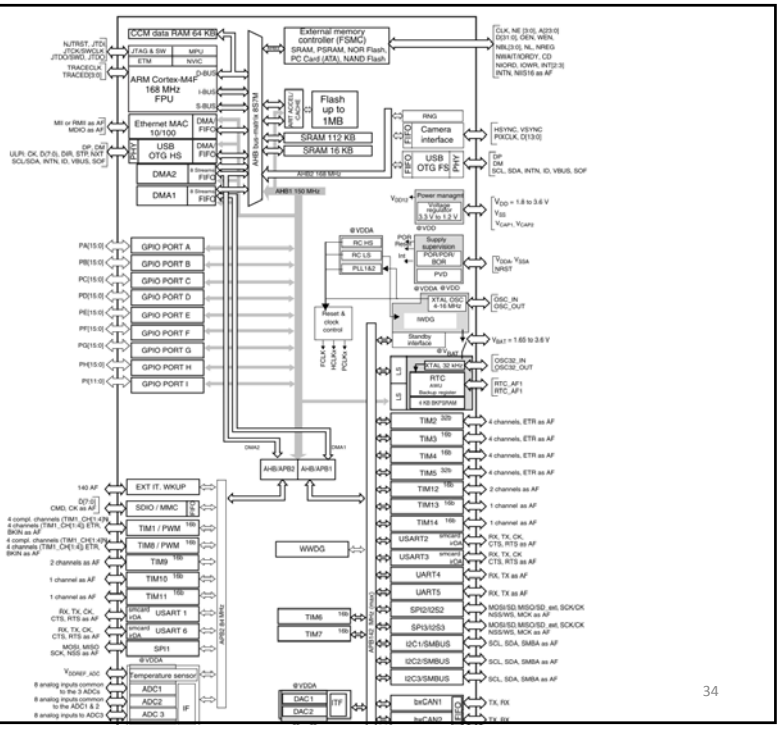
Exception Handling (System Call)



32


SWD
M8997A
www.st.com/stm32f4-discovery


Mini USB
Embedded ST-LINK/V2
SWD
STM32F407VGT6
I/O
I/O
I/O
RESET
Header
Header
LED LD3 to LD8
B2 RST
MP45DT02
B1 USER
CS43L22
LIS302DL
Mini-Jack
Micro-USB


CCM data RAM 64 KB
External memory controller (FSMC)
ARM Cortex-M4F 168 MHz FPU
Flash up to 1MB
SRAM 112 KB
SRAM 16 KB
Ethernet MAC 10/100
USB OTG HS
USB OTG FS
Camera interface
RNG
DMA1
DMA2
GPIO PORT A
GPIO PORT B
GPIO PORT C
GPIO PORT D
GPIO PORT E
GPIO PORT F
GPIO PORT G
GPIO PORT H
GPIO PORT I
RTC
WWDG
USART1
USART2
USART3
UART4
UART5
SPI1
SPI2/I2S2
SPI3/I2S3
I2C1/SMBUS
I2C2/SMBUS
I2C3/SMBUS
SDIO / MMC
Temperature sensor
ADC1
ADC2
ADC 3
DAC1
DAC2
bxCAN1
bxCAN2
TIM1 / PWM
TIM2
TIM3
TIM4
TIM5
TIM6
TIM7
TIM8 / PWM
TIM9
TIM10
TIM11
TIM12
TIM13
TIM14

Bus	Boundary address	Peripheral
AHB1	0x4004 0000 - 0x4007 FFFF	USB OTG HS
	0x4002 9400 - 0x4003 FFFF	Reserved
	0x4002 9000 - 0x4002 93FF	ETHERNET MAC
	0x4002 8C00 - 0x4002 8FFF	
	0x4002 8800 - 0x4002 8BFF	
	0x4002 8400 - 0x4002 87FF	
	0x4002 8000 - 0x4002 83FF	
	0x4002 6800 - 0x4002 7FFF	Reserved
	0x4002 6400 - 0x4002 67FF	DMA2
	0x4002 6000 - 0x4002 63FF	DMA1
	0x4002 5000 - 0x4002 5FFF	Reserved
	0x4002 4000 - 0x4002 4FFF	BKPSRAM
	0x4002 3C00 - 0x4002 3FFF	Flash interface register
	0x4002 3800 - 0x4002 3BFF	RCC
	0x4002 3400 - 0x4002 37FF	Reserved
	0x4002 3000 - 0x4002 33FF	CRC
	0x4002 2400 - 0x4002 2FFF	Reserved
	0x4002 2000 - 0x4002 23FF	GPIOI
	0x4002 1C00 - 0x4002 1FFF	GPIOH
	0x4002 1800 - 0x4002 1BFF	GPIOG
	0x4002 1400 - 0x4002 17FF	GPIOF
	0x4002 1000 - 0x4002 13FF	GPIOE
	0x4002 0C00 - 0x4002 0FFF	GPIOD
	0x4002 0800 - 0x4002 0BFF	GPIOC
	0x4002 0400 - 0x4002 07FF	GPIOB
	0x4002 0000 - 0x4002 03FF	GPIOA
	0x4001 5800- 0x4001 FFFF	Reserved

35

GPIO resources

- Each general-purpose I/O port has
  - four 32-bit configuration registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR and GPIOx_PUPDR)
  - two 32-bit data registers (GPIOx_IDR and GPIOx_ODR)
  - a 32-bit set/reset register (GPIOx_BSRR)
  - a 32-bit locking register (GPIOx_LCKR)
  - two 32-bit alternate function selection register (GPIOx_AFRH and GPIOx_AFRL)

36

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]	PUPDR(i) [1:0]		I/O configuration	
01	0	SPEED [B:A]	0	0	GP output	PP
	0		0	1	GP output	PP + PU
	0		1	0	GP output	PP + PD
	0		1	1	Reserved	
	1		0	0	GP output	OD
	1		0	1	GP output	OD + PU
	1		1	0	GP output	OD + PD
	1		1	1	Reserved (GP output OD)	
10	0	SPEED [B:A]	0	0	AF	PP
	0		0	1	AF	PP + PU
	0		1	0	AF	PP + PD
	0		1	1	Reserved	
	1		0	0	AF	OD
	1		0	1	AF	OD + PU
	1		1	0	AF	OD + PD
	1		1	1	Reserved	
00	x	x x	0	0	Input	Floating
	x	x x	0	1	Input	PU
	x	x x	1	0	Input	PD
	x	x x	1	1	Reserved (input floating)	
11	x	x x	0	0	Input/output	Analog
	x	x x	0	1	Reserved	
	x	x x	1	0		
	x	x x	1	1		

37

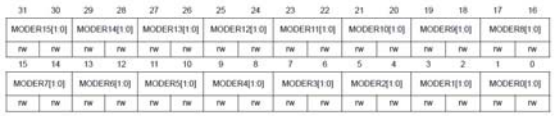
- #define GPIOD_BASE (AHB1PERIPH_BASE + 0x0C00)
- #define AHB1PERIPH_BASE (PERIPH_BASE + 0x00020000)
- #define PERIPH_BASE ((uint32_t)0x40000000)

38

```
typedef struct
{
  __IO uint32_t MODER;    /*!< GPIO port mode register,               Address
offset: 0x00      */
  __IO uint32_t OTYPER;   /*!< GPIO port output type register,        Address
offset: 0x04      */
  __IO uint32_t OSPEEDR;  /*!< GPIO port output speed register,       Address
offset: 0x08      */
  __IO uint32_t PUPDR;    /*!< GPIO port pull-up/pull-down register,  Address
offset: 0x0C      */
  __IO uint32_t IDR;      /*!< GPIO port input data register,         Address
offset: 0x10      */
  __IO uint32_t ODR;      /*!< GPIO port output data register,        Address
offset: 0x14      */
  __IO uint16_t BSRRL;    /*!< GPIO port bit set/reset low register,  Address
offset: 0x18      */
  __IO uint16_t BSRRH;    /*!< GPIO port bit set/reset high register, Address
offset: 0x1A      */
  __IO uint32_t LCKR;     /*!< GPIO port configuration lock register, Address
offset: 0x1C      */
  __IO uint32_t AFR[2];   /*!< GPIO alternate function registers,     Address
offset: 0x20-0x24 */
} GPIO_TypeDef;
```

39

- GPIOx->MODER: GPIOx bus pin mode
  - 0x55000000

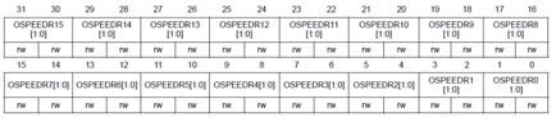


31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 2y:2y+1 **MODERy[1:0]:** Port x configuration bits (y = 0..15)
These bits are written by software to configure the I/O direction mode.
00: Input (reset state)
01: General purpose output mode
10: Alternate function mode
11: Analog mode

40

- GPIOx->OSPEEDR: GPIOx bus speed
  - High speed (100MHz)
  - 0xFF000000



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]		OSPEEDR14 [1:0]		OSPEEDR13 [1:0]		OSPEEDR12 [1:0]		OSPEEDR11 [1:0]		OSPEEDR10 [1:0]		OSPEEDR9 [1:0]		OSPEEDR8 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1 [1:0]		OSPEEDR0 1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 2y:2y+1 **OSPEEDRy[1:0]:** Port x configuration bits (y = 0..15)
These bits are written by software to configure the I/O output speed.
00: Low speed
01: Medium speed
10: Fast speed
11: High speed
Note: Refer to the product datasheets for the values of OSPEEDRy bits versus V_{DD} range and external load.

41