



มหาวิทยาลัย
ราชภัฏนครปฐม



บทเรียนออนไลน์

รายวิชา 6562213 ไมโครโปรเซสเซอร์ 1
6562213 Microprocessor 1

โดย

ผู้ช่วยศาสตราจารย์ วีระ กาญจนสินธุ์
วท.ม.(ฟิสิกส์) วท.บ.(ฟิสิกส์)
คณะวิทยาศาสตร์และเทคโนโลยี

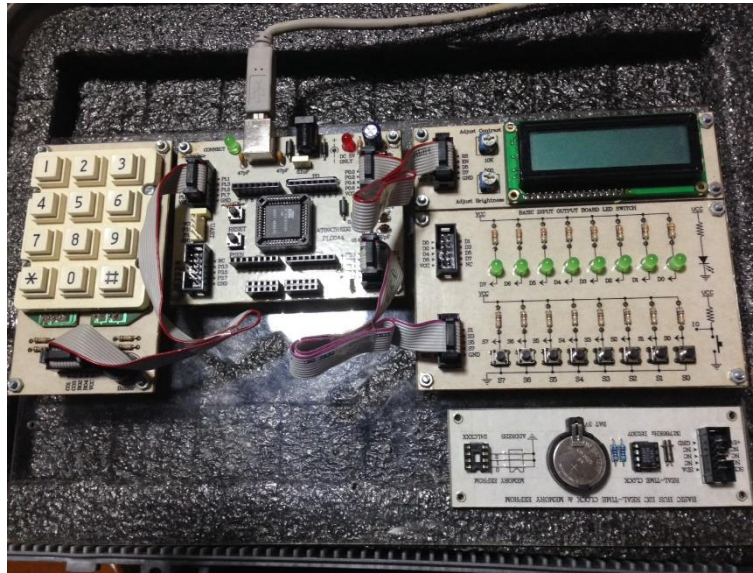


จุดประสงค์การเรียนรู้ของรายวิชา 6562213 ไมโครโพรเซสเซอร์ 1



2) เพื่อให้ผู้เรียนสามารถ
พัฒนาซอฟต์แวร์ภาษาซี
ในการควบคุมการทำงานของ
ฮาร์ดแวร์ที่ต้องวงจรร่วมกับ
ชิปตระกูล MCS-51

1) เพื่อให้ผู้เรียนสามารถ
ออกแบบและพัฒนาฮาร์ดแวร์
ที่มีไมโครคอนโทรลเลอร์ หรือ
ชิปตระกูล MCS-51
เป็นหน่วยประมวลผล



3) เพื่อให้ผู้เรียนสามารถ
พัฒนาเจตคติและมีจรรยาบรรณ
ในการคิดค้นและพัฒนา
สิ่งประดิษฐ์ โดยใช้ชิปตระกูล
MCS-51 เป็นหน่วยประมวลผล





- **บทที่ 1** สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน
- **บทที่ 2** โครงสร้างภายในของไมโครคอนโทรลเลอร์
- **บทที่ 3** ชุดคำสั่งแยกตามประเภทการใช้งาน
- **บทที่ 4** ตัวอย่างการเขียนโปรแกรมเบื้องต้นและโปรแกรมใช้งาน
- **บทที่ 5** การออกแบบวงจรหน่วยความจำและอุปกรณ์ต่อร่วม
- **บทที่ 6** การประยุกต์ใช้งานไมโครคอนโทรลเลอร์



มหาวิทยาลัย
ราชภัฏนครปฐม



รายวิชา 6562213 ไมโครโพรเซสเซอร์ 1

บทที่ 1 สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน

โดยผู้ช่วยศาสตราจารย์ธีระ กาญจนสินธุ์

คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยราชภัฏนครปฐม



บทที่ 1 สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน

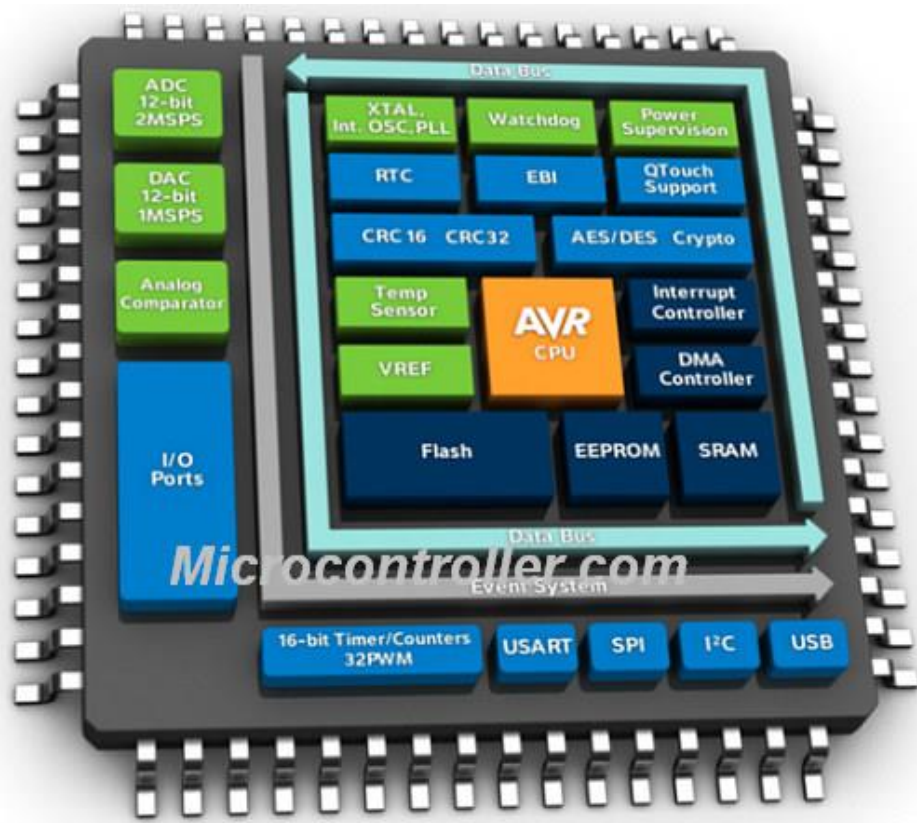


บทนำ

ในบทนี้ มีเนื้อหาเกี่ยวกับความแตกต่างระหว่าง CPU กับ MCU **MCU** นับเป็นคอมพิวเตอร์ในรูปชิปไอซี ที่ถูกสร้างขึ้นมาเพื่อใช้ในงานควบคุมที่เฉพาะเจาะจง จึงมีชื่อเรียกว่า **computer in a single chip**

ปัจจุบันมี MCU จากบริษัทผู้ผลิตที่หลากหลาย ซึ่งต่างมีสมรรถนะที่รองรับความต้องการของ **ผู้ใช้งาน** ในด้านความเร็ว-ประหยัดพลังงาน-ช่องทางสื่อสาร-รองรับอุปกรณ์รอบ

เทคโนโลยีที่ก้าวไปสู่ระบบ **สมองกลฝังตัว** นับว่ามีรากฐานที่ **พัฒนามาจาก MCU**

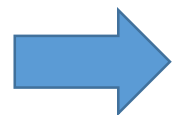




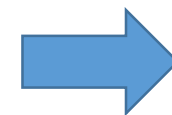
หัวข้อสำคัญในบทที่ 1



1.1 ทบทวน
บิต-นินเบิล-
ไบต์-เวิร์ด/
MSB-LSB/
BIN2HEX



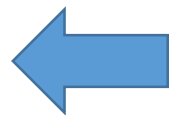
1.2
ไมโครโพรเซสเซอร์
VS
ไมโครคอนโทรลเลอร์



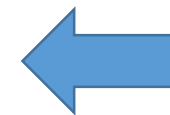
1.3 ตระกูลของ
ไมโครคอนโทรลเลอร์
และความแตกต่าง



1.6 สถาปัตยกรรมของ
MCU ตระกูล MCS-51



1.5 การพัฒนาฮาร์ดแวร์
และซอฟต์แวร์บน MCU



1.4 ภาษาที่ใช้
ในการพัฒนา
โปรแกรมควบคุม



บทที่ 1 สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน



1.1 ทบทวน บิต - นิบเบิล - ไบต์ - เวิร์ด - ดับเบิลเวิร์ด



บิต (bit)

คือ...ข้อมูลฐานสองขนาด 1 หลัก เช่น... **1** หรือ **0**



นิบเบิล (nibble)

คือ...ข้อมูลฐานสองขนาด 4 หลัก เช่น... **1001** หรือ **0110**

ไบต์ (byte)

คือ...ข้อมูลฐานสองขนาด 8 หลัก
เช่น... **1001 1110** หรือ **1011 0110**

เวิร์ด (word)

คือ...ข้อมูลฐานสองขนาด 16 หลัก
เช่น... **1001 1110 1011 0110** หรือ **1101 1010 1111 0010**

ดับเบิลเวิร์ด (double word)

คือ...ข้อมูลฐานสองขนาด 32 หลัก
เช่น... **1001 1110 1011 0110 1101 1010 1111 0010**



บทที่ 1 สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน

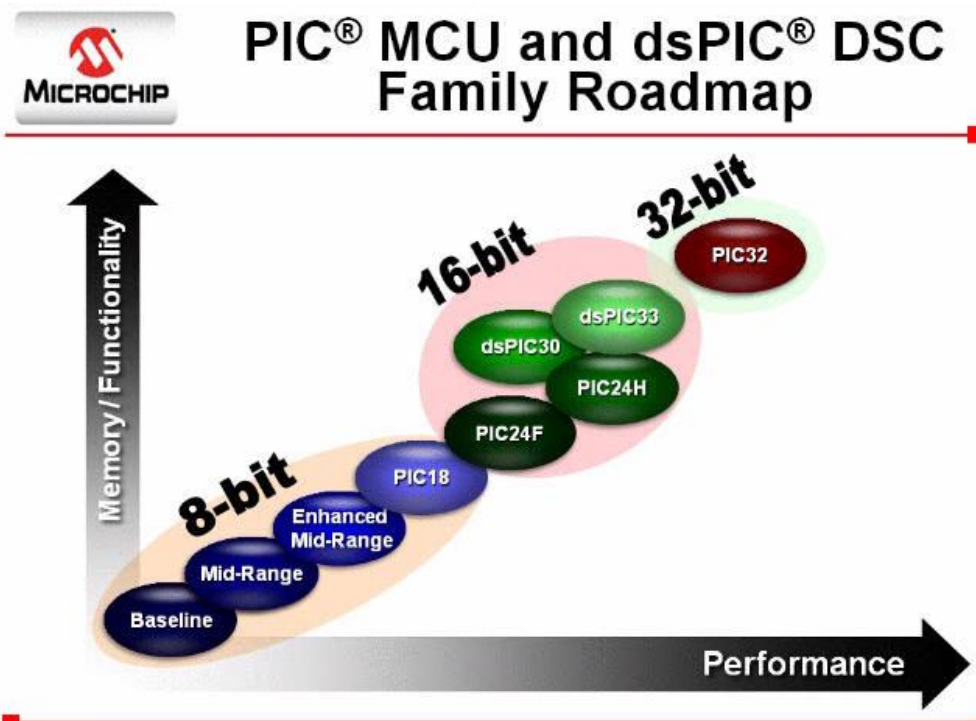


1.1 ทบทวน บิต - นิบเบิล - ไบต์ - เวิร์ด - ดับเบิลเวิร์ด

พัฒนาการของ ชิปไมโครคอนโทรลเลอร์ (MCU)



ความเข้าใจในนิยามของ บิต-นิบเบิล-ไบต์-ดับเบิลเวิร์ด
นำไปสู่ความเข้าใจถึง...พัฒนาการของ ชิปไมโครคอนโทรลเลอร์
เพื่อรองรับงานนวัตกรรมที่อำนวยความสะดวกมากขึ้น



- ประหยัดพลังงาน
- รองรับการใช้เครือข่าย
- รองรับการเชื่อมต่อ
- รองรับความนิรภัย
- เน้นคุณสมบัติกำลัง
- รองรับระบบสัมผัส
- ความเร็วประมวลผล
- ช่องทางสื่อสาร



บทที่ 1 สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน

1.1 ทบทวน บิต - นิบเบิล - ไบต์ - เวิร์ด - ดับเบิลเวิร์ด



ทบทวน บิตนัยสำคัญสูงสุด (MSB: Most Significant Bit)
บิตนัยสำคัญต่ำสุด (LSB: Least Significant Bit)

ตัวอย่างข้อมูลฐานสองขนาด 20 บิต

1001 1101 1010 0111 1110



บิตสุดท้ายซ้ายมือสุด เรียกว่า MSB

บิตสุดท้ายขวามือสุด เรียกว่า LSB



บทที่ 1 สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน



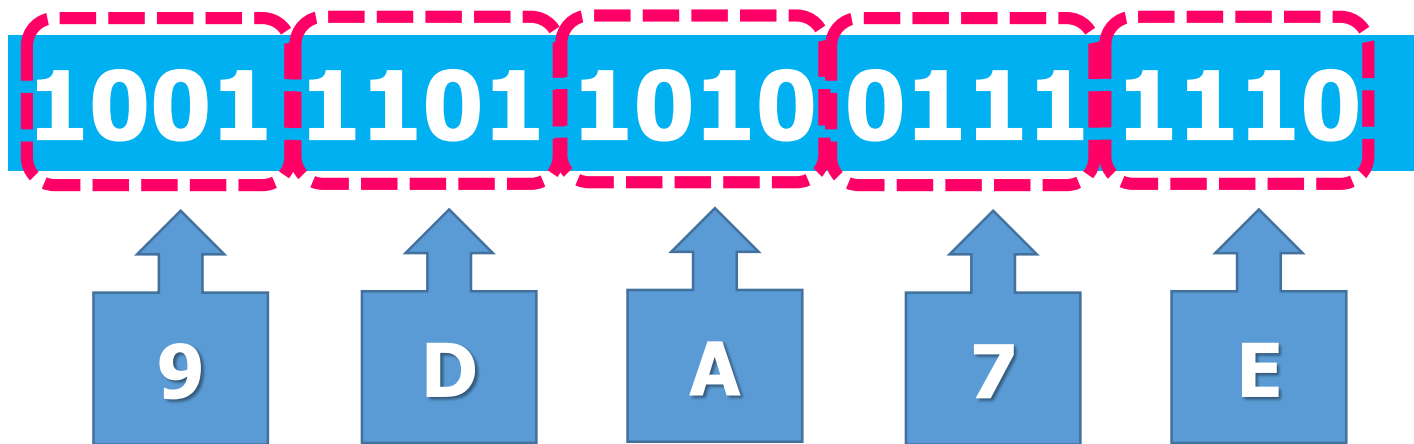
1.1 ทบทวน บิต - นิบเบิล - ไบต์ - เวิร์ด - ดับเบิลเวิร์ด



ทบทวน การอ่านในรูปเลขฐานสิบหก
โดยอ่านเลขฐานสอง...จากขวามือไปซ้ายมือ ทีละ 4 บิต
ช่วยให้คนเราสื่อสารกันง่ายกว่า การอ่านในรูปเลขฐานสอง

ตัวอย่างข้อมูลฐานสองขนาด 20 บิต

อ่านเป็นฐานสิบหก จากขวาไปซ้าย



$$10011101101001111110_2 = 9DA7E_{16}$$



บทที่ 1 สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน

1.2 ไมโครโพรเซสเซอร์ VS ไมโครคอนโทรลเลอร์



นิยามของไมโครโพรเซสเซอร์ (CPU)

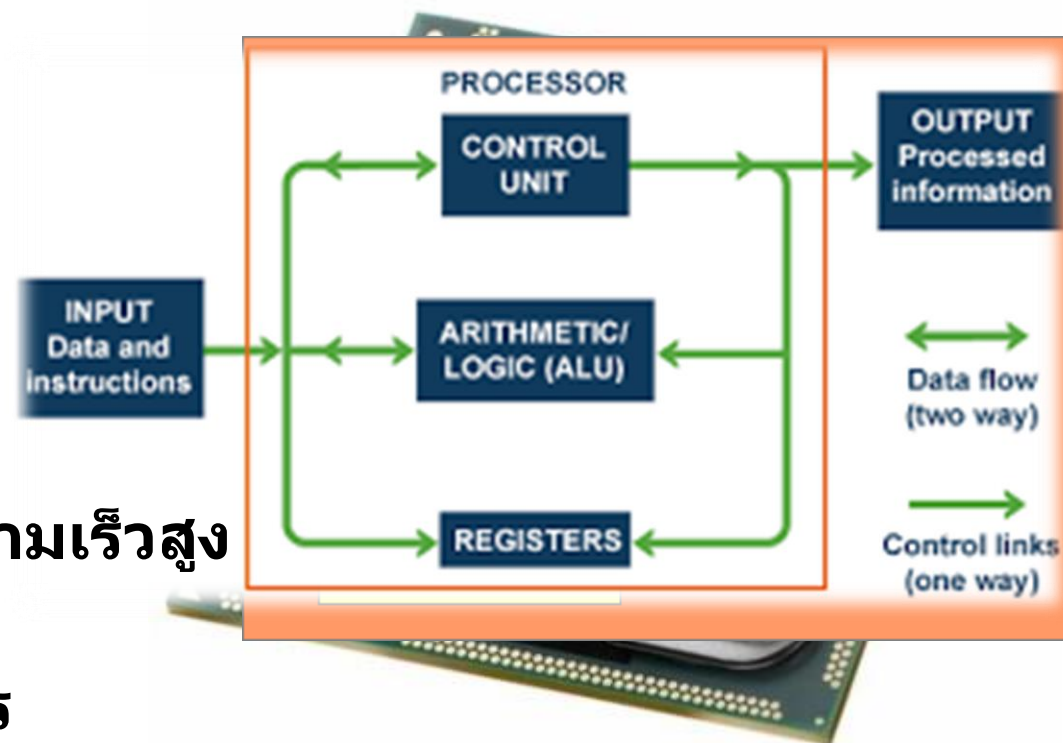
Microprocessor (CPU)

คือ...หน่วยประมวลผลกลาง หรือ
CPU (Central Processing Unit)

เป็น...หน่วยหลักในคอมพิวเตอร์

ทำหน้าที่ควบคุมการทำงานของอุปกรณ์/
และประมวลผลเลขคณิตและตรรกะ/

สามารถประมวลผลคำสั่งและคำนวณข้อมูลด้วยความเร็วสูง
มีโครงสร้างภายในที่เกี่ยวข้องกับเรจิสเตอร์/บัส/
หน่วยความจำ/ วงจรนับ/ วงจรจับเวลา/ ช่องสื่อสาร





บทที่ 1 สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน

1.2 ไมโครโพรเซสเซอร์ VS ไมโครคอนโทรลเลอร์

นิยามของไมโครคอนโทรลเลอร์ (MCU)

Microcontroller (MCU)

คือ...คอมพิวเตอร์ขนาดเล็กเท่าชิปไอซี

ภายในชิปไมโครคอนโทรลเลอร์
มีโครงสร้างเหมือนคอมพิวเตอร์

โดยประกอบด้วย

1) หน่วยประมวลผล

2) หน่วยความจำ

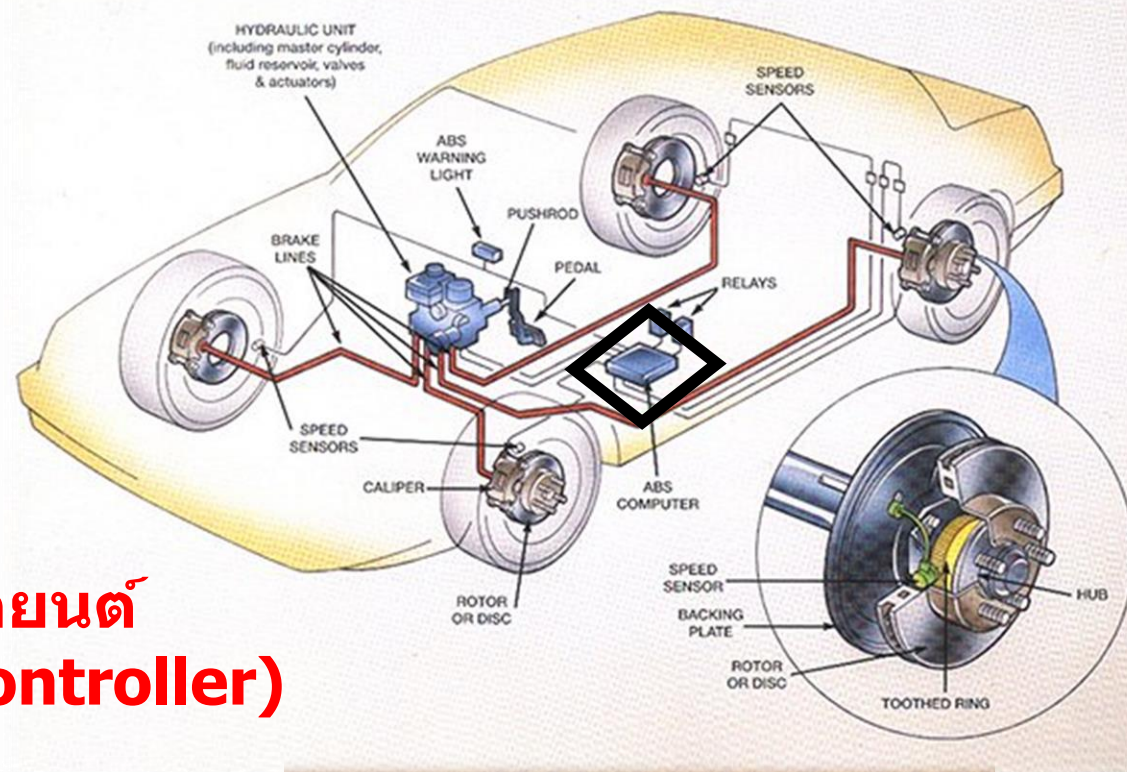
3) หน่วยรับ/ส่งข้อมูล

นิยมใช้ในงานควบคุมเครื่องจักรกล

ตัวอย่างเช่น ระบบขับเคลื่อนของรถยนต์
ที่ควบคุมโดยกล่อง MCU (microcontroller)

External Interrupts

Counter Input





บทที่ 1 สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน



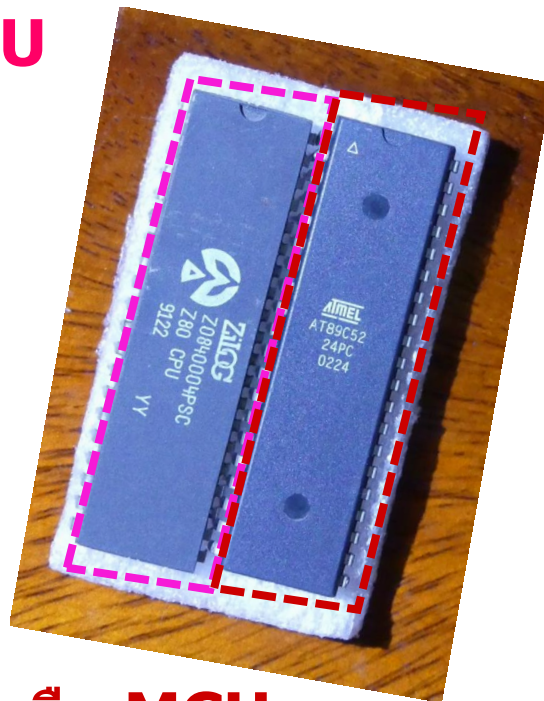
1.2 ไมโครโพรเซสเซอร์ VS ไมโครคอนโทรลเลอร์

ความแตกต่าง



Microprocessor (CPU)

คือ...หน่วยประมวลผลกลาง
คือ...CPU

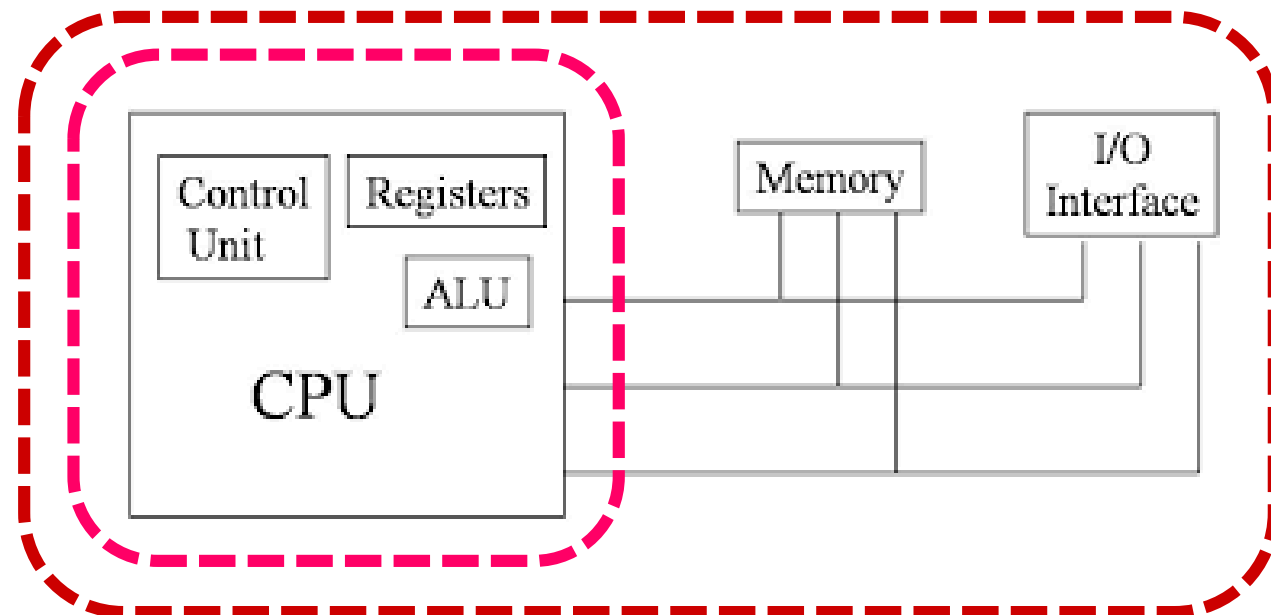


รูปซ้ายมือ คือ CPU
...CPU : Z-80

รูปขวามือ คือ MCU
...microcontroller : AT89C51

Microcontroller (MCU)

คือ...คอมพิวเตอร์ขนาดเล็กเท่าชิปไอซี
คือ...CPU + memory + I/O = computer





1.2 ไมโครโพรเซสเซอร์ VS ไมโครคอนโทรลเลอร์



Microprocessor (CPU)

ไม่มี ROM/ RAM ภายในชิป
มีเพียง CPU

ตัวอย่าง

Pentium (Intel)
PowerPC (Motorola)
Spark (SUN)
Z80 (Zilog)

การใช้งาน

Desktop/ laptop/
workstations/ Server/

General Purpose

Microcontroller (MCU)

มี ROM/ RAM ภายในชิป
มี CPU+ memory+ I/O อยู่ในชิปเดียวกัน

8051 (Intel)
68HC08 (Motorola)
AVR (Atmel)
Z8 (Zilog)
PIC (Microchip)
ARM (Advance RISC Machine)

TV remote control/ video games/
Robot/

Specific Purpose



1.3 ตระกูลของไมโครคอนโทรลเลอร์และความแตกต่าง



Microcontroller Families

คือ...ไมโครคอนโทรลเลอร์ ที่ผลิตจากหลากหลายบริษัทผู้ผลิตชั้นนำในยุโรป เอเชีย และอเมริกา

ALTERA	LAN MICROELECTRONICS	Infineon
ANALOG DEVICES	EPSON EXCEED YOUR VISION	Intel
Atmel	freescale semiconductor	LA SEMIC
CYPRESS EMBEDDED IN TOMORROW™	FUJITSU	MICROCHIP
maxim integrated	HOLTEK	National Semiconductor



ความแตกต่าง

ประหยัดพลังงาน

รองรับการใช้เครือข่าย

รองรับการเชื่อมต่อ

รองรับความนิรภัย

เน้นคุณสมบัติกำลัง

รองรับระบบสัมผัส

ช่องทางสื่อสาร

ความแตกต่างขึ้นกับ market strategy plan ของบริษัทผู้ผลิต



1.3 ตระกูลของไมโครคอนโทรลเลอร์และความแตกต่าง

Microcontroller Families

คือ...ตระกูลไมโครคอนโทรลเลอร์ (MCU) ที่บริษัทผู้ผลิตทั่วโลก
ต่างอ้างอิงและใช้เทคโนโลยีการผลิต มี 4 ตระกูลหลัก
ได้แก่... 1) MCS-51 2) PIC 3) AVR 4) ARM



MCS-51:
MiCrocontrollerS-51

89S52
89S51
89C52
89C51
89V51
.....

PIC:
Peripheral Interface
Controller

PIC10Fxx
PIC12Fxx
PIC16Fxx
.....

AVR:
Alf and Vegard's
RISC processor

ATmega8
ATmega16
ATmega32
(ATtiny/
ATmega/
Atxmega)

ARM:
Advanced RISC
Machine

LPC2148
ARM Cortex-M0
ARM Cortex-M1
ARM Cortex-M2
.....
ARM Cortex-M6
ARM Cortex-M7

ความแตกต่าง

ประหยัดพลังงาน

รองรับการใช้เครือข่าย

รองรับการเชื่อมต่อ

รองรับความนิรภัย

เน้นคุณสมบัติกำลัง

รองรับระบบสัมผัส

เน้นความปลอดภัย

ช่องทางสื่อสาร



1.3 ตระกูลของไมโครคอนโทรลเลอร์และความแตกต่าง



ความแตกต่าง

ประหยัดพลังงาน

รองรับการใช้เครือข่าย

รองรับการเชื่อมต่อ

รองรับความนิรภัย

เน้นคุณสมบัติกำลัง

รองรับระบบสัมผัส

เน้นความปลอดภัย

RISC

Reduced Instruction Set Computing

ชุดคำสั่งขนาดเล็กกว่าและจำนวนน้อยกว่า
มีคำสั่งเกี่ยวกับ addressing mode น้อยกว่า
ขนาดคำสั่งคงที่ (32 บิต) ใช้ 1 m/c cycle

สมรรถนะการทำงานเน้นที่..ซอฟต์แวร์
ฮาร์ดแวร์ไม่ซับซ้อน ต้นทุนการผลิตไม่สูง
ใช้พลังงานไฟฟ้าไม่สูง

การทำงานต่อเนื่องอาจใช้กระบวนการ pipelining
การปรับความเร็วให้สูงขึ้นใช้ชิปน้อยกว่า
ตัวอย่างชิป: PIC & ARM

ตัวอย่างการใช้งาน: Smartphones, PDAs

CISC

Complex Instruction Set Computing

ชุดคำสั่งมากกว่าและซับซ้อนกว่า
มีคำสั่งเกี่ยวกับ addressing mode มากกว่า
ขนาดคำสั่งใช้ 2-10 m/c cycle

สมรรถนะการทำงานเน้นที่..ฮาร์ดแวร์
ฮาร์ดแวร์ซับซ้อน ต้นทุนการผลิตสูงกว่า
ใช้พลังงานไฟฟ้าสูง

การทำงานต่อเนื่องใช้ pipelining ไม่ได้
การปรับความเร็วให้สูงขึ้นใช้ชิปจำนวนมาก
ตัวอย่างชิป: x86 & 8051

ตัวอย่างการใช้งาน: Security systems,
Home automation.

ที่มา: <https://www.technobyte.org/difference-between-risc-and-cisc/>



1.3 ตระกูลของไมโครคอนโทรลเลอร์และความแตกต่าง



Microcontroller Families

..ช่องสื่อสาร: UART/ USART/ SPI/ I²C

8051 Microcontroller:

Bus width: 8-bit for standard coreขนาดบัส : 8 บิต

Communication Protocols: UART, USART, SPI, I2C

Speed: 12 Clock/instruction cycleความเร็ว: 1 คำสั่ง/12 clk.

Memory: ROM, SRAM, FLASH ..ชนิดหน่วยจำ : ROM, SRAM/Flash

ISA: CLSC

Memory Architecture: Von Neumann architecture

Power Consumption: Average

Families: 8051 variants ..ตระกูล: 8051

Popular Microcontrollers: AT89C51, P89v51, etc.

ความแตกต่าง

ประหยัดพลังงาน

รองรับการใช้เครือข่าย

รองรับการเชื่อมต่อ

รองรับความนิรภัย

เน้นคุณสมบัติกำลัง

รองรับระบบสัมผัส

เน้นความปลอดภัย



1.3 ตระกูลของไมโครคอนโทรลเลอร์และความแตกต่าง



Microcontroller Families ..ช่องสื่อสาร: PIC/UART/LIN/Ethernet/SPI

PIC Microcontroller:

Bus width: 8/16/32-bit

.....ขนาดบัส : 8/ 16/ 32 บิต

Communication Protocols: PIC, UART, USART, LIN, CAN, Ethernet, SPI, I2S

Speed: 4 Clock/instruction cycle

.....ความเร็ว: 1 คำสั่ง/4 clk.

Memory: SRAM, FLASH

.....ชนิดหน่วยจำ : SRAM/Flash

ISA: Some feature of RISC

Memory Architecture: Harvard architecture

Power Consumption: Low

Families: PIC16, PIC17, PIC18, PIC24, PIC32

..ตระกูล: PIC

Popular Microcontrollers: PIC18fXX8, PIC16f88X, PIC32MXX

ความแตกต่าง

ประหยัดพลังงาน

รองรับการใช้เครือข่าย

รองรับการเชื่อมต่อ

รองรับความนิรภัย

เน้นคุณสมบัติกำลัง

รองรับระบบสัมผัส

เน้นความปลอดภัย



1.3 ตระกูลของไมโครคอนโทรลเลอร์และความแตกต่าง



Microcontroller Families ของสื่อสาร: UART/ SPI/ I²C/ USB/ Ethernet

AVR Microcontroller:

Bus width: 8/32-bit

.....ขนาดบัส : 8/ 32 บิต

Communication Protocols: UART, USART, SPI, I2C, (special purpose AVR support CAN, USB, Ethernet)

Speed: 1 clock/ instruction cycle

.....ความเร็ว: 1 คำสั่ง/1 clk.

Memory: Flash, SRAM, EEPROM

หน่วยจำ : Flash/SRAM E²PROM

ISA: RISC

Memory Architecture: Modified

Power Consumption: Low

Families: Tiny, Atmega, Xmega, special purpose AVR ..ตระกูล: AVR

Popular Microcontrollers: Atmega8, 16, 32, Arduino Community

ความแตกต่าง

ประหยัดพลังงาน

รองรับการใช้เครือข่าย

รองรับการเชื่อมต่อ

รองรับความนิรภัย

เน้นคุณสมบัติกำลัง

รองรับระบบสัมผัส

เน้นความปลอดภัย



1.3 ตระกูลของไมโครคอนโทรลเลอร์และความแตกต่าง



Microcontroller Families

...ช่องสื่อสาร: **UART/ SPI/ I²C/ USB/ IrDA**

ARM Microcontroller:

Bus width: 32-bit mostly also available in 64-bit ...ขนาดบัส : **32 บิต**

Communication Protocols: UART, USART, LIN, I2C, SPI, CAN, USB, Ethernet, I2S, DSP, SAI (serial audio interface), IrDA

Speed: 1 clock/ instruction cycleความเร็ว: **1 คำสั่ง/1 clk.**

Memory: Flash, SDRAM, EEPROM หน่วยจำ : **Flash/SDRAM E²PROM**

ISA: RISC

Memory Architecture: Modified Harvard architecture

Power Consumption: Low

Families: ARMv4,5,6,7 and series

Popular Microcontrollers: LPC2148, ARM Cortex-M0 to ARM Cortex-M7, etc.

ความแตกต่าง

ประหยัดพลังงาน

รองรับการใช้เครือข่าย

รองรับการเชื่อมต่อ

รองรับความนิรภัย

เน้นคุณสมบัติกำลัง

รองรับระบบสัมผัส

เน้นความปลอดภัย



1.4 ภาษาที่ใช้ในการพัฒนาโปรแกรมควบคุม

คือ...สื่อภาษาที่มนุษย์เรา ใช้เพื่อสื่อสารรับ-ส่ง ข้อมูล-คำสั่ง
...ระหว่างชิป MCU กับอุปกรณ์รายรอบในงานเฉพาะเจาะจง
...เช่น การควบคุมการติด-ดับของไฟ แสงสว่าง/
.....การควบคุมการหมุน-หยุดหมุนของมอเตอร์
.....การทำงานของอุปกรณ์แบบวัฏจักร เช่น ไฟจราจร/ ไฟประดับ
.....การทำงานของอุปกรณ์แบบมีเงื่อนไข เช่น ลิฟต์ (elevator)
.....สายพานลำเลียง (conveyor)





1.4 ภาษาที่ใช้ในการพัฒนาโปรแกรมควบคุม

ภาษาที่ใช้ในการพัฒนาโปรแกรมควบคุมอุปกรณ์
(ภาษาระดับสูง-ภาษาแอสเซมบลี-ภาษาเครื่อง)
บนชิปไมโครคอนโทรลเลอร์



ภาษาระดับสูง

FORTRAN

C

Pascal

High-Level Language

Assembly Language

Machine Language

Hardware

ภาษาแอสเซมบลี

ภาษาเครื่อง

ตัวอย่างภาษาเครื่อง

Example of machine-language

Here's what a program-fragment looks like:

```
10100001 10111100 10010011 00000100
00001000 00000011 00000101 11000000
10010011 00000100 00001000 10100011
11000000 10010100 00000100 00001000
```

It means: $z = x + y;$



1.4 ภาษาที่ใช้ในการพัฒนาโปรแกรมควบคุม

ภาษาที่ใช้ในการพัฒนาโปรแกรมควบคุมอุปกรณ์
(ภาษาระดับสูง-ภาษาแอสเซมบลี-ภาษาเครื่อง)
บนชิปไมโครคอนโทรลเลอร์



ภาษาระดับสูง

FORTRAN

C

Pascal

High-Level Language

Assembly Language

Machine Language

Hardware

ภาษาแอสเซมบลี

ภาษาเครื่อง

ตัวอย่างภาษาแอสเซมบลี

```
1 ORG 0000H
2 MOV DPTR, #200H
3 CLR A
4 MOVC A, @A+DPTR
5 MOV R0, A
6
7 MOV DPTR, #300H
8 CLR A
9 MOVC A, @A+DPTR
10 MOV R1, A
11
12 //compare R0 to R1, compare the rest of the array characters
13
14 Here: SJMP HERE
15
16
17 ORG 200H
18 MYDATA1: DB "abc"
19
20 ORG 300H
21 MYDATA2: DB "ab,abc,abcd."
22 END
```



1.4 ภาษาที่ใช้ในการพัฒนาโปรแกรมควบคุม

ภาษาที่ใช้ในการพัฒนาโปรแกรมควบคุมอุปกรณ์
(ภาษาระดับสูง-ภาษาแอสเซมบลี-ภาษาเครื่อง)
บนชิปไมโครคอนโทรลเลอร์



High-level Language
ภาษาระดับสูง

temp = v[k];
v[k+1] = temp;
FORTRAN

TEMP = V(K)
V(K+1) = TEMP
Pascal

Java Compiler
High-Level Language
Fortran Compiler

Assembly Language
ภาษาแอสเซมบลี

lw St1, 4(S2)
sw St0, 4(S2)
Assembly Language

MIPS Assembler
Machine Language

Machine Language
ภาษาเครื่อง

0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
Hardware

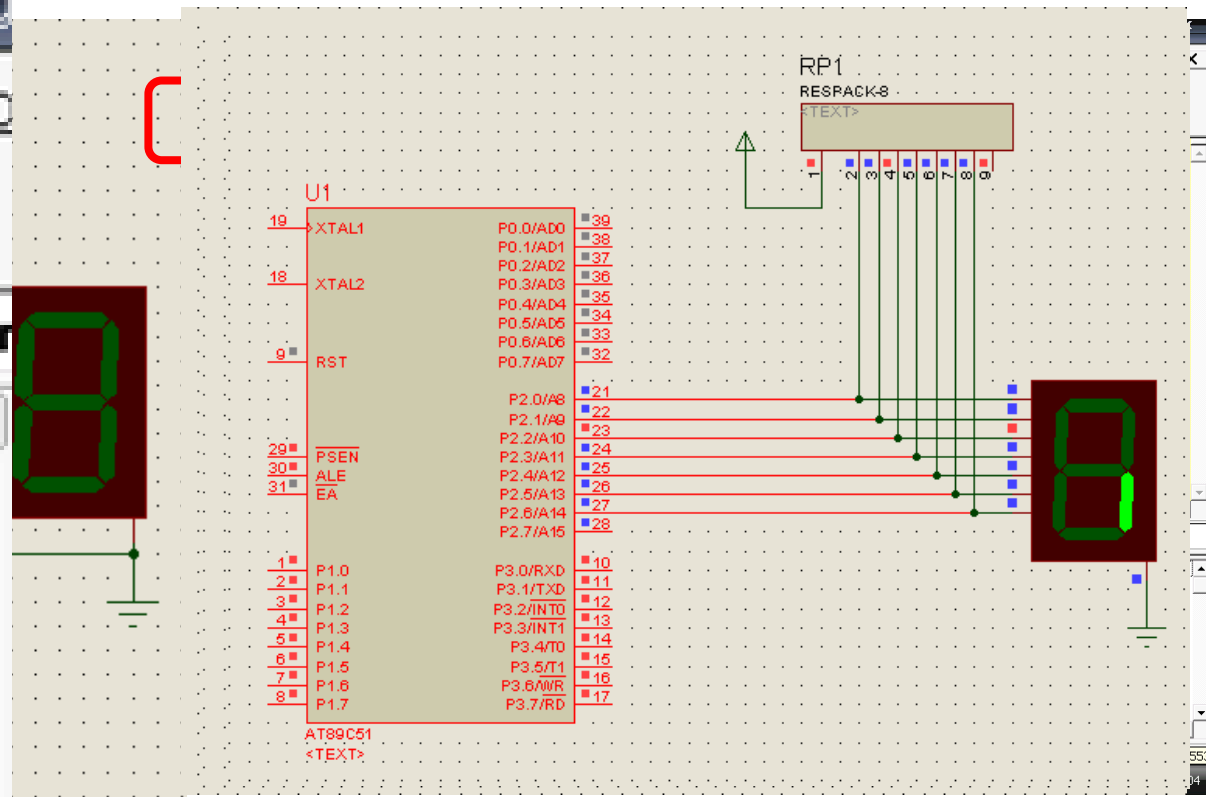
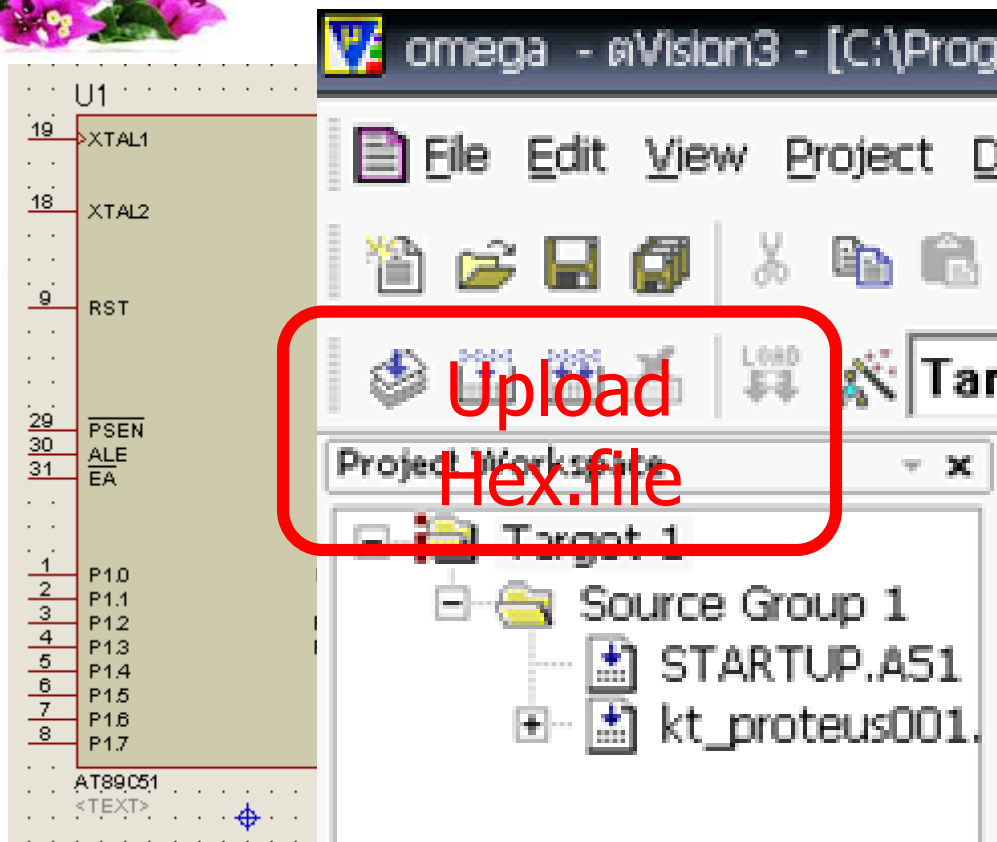
ตัวอย่างภาษาซี

```
01 #include<reg51.H>
02 void delay(unsigned int num)
03 {unsigned int i,j;
04     for(i=0;i<num;i++)
05         {for(j=0;j<10000;j++);
06         }
07 }
08 void main(void)
09 { while(1)
10     { P2=0X01;delay(10);
11       P2=0X02;delay(10);
12       P2=0X04;delay(10);
13     }
14 }
15
```




บทที่ 1 สถาปัตยกรรมไมโครโปรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน

1.5 การพัฒนาฮาร์ดแวร์และซอฟต์แวร์บน MCU



ขั้นที่ 1
ออกแบบวงจร
circuit design

ขั้นที่ 2
ประกอบวงจร
circuit assembly

ขั้นที่ 3
พัฒนาซอฟต์แวร์
C-editor

ขั้นที่ 4
uploadHex. File
C2Hex.complier

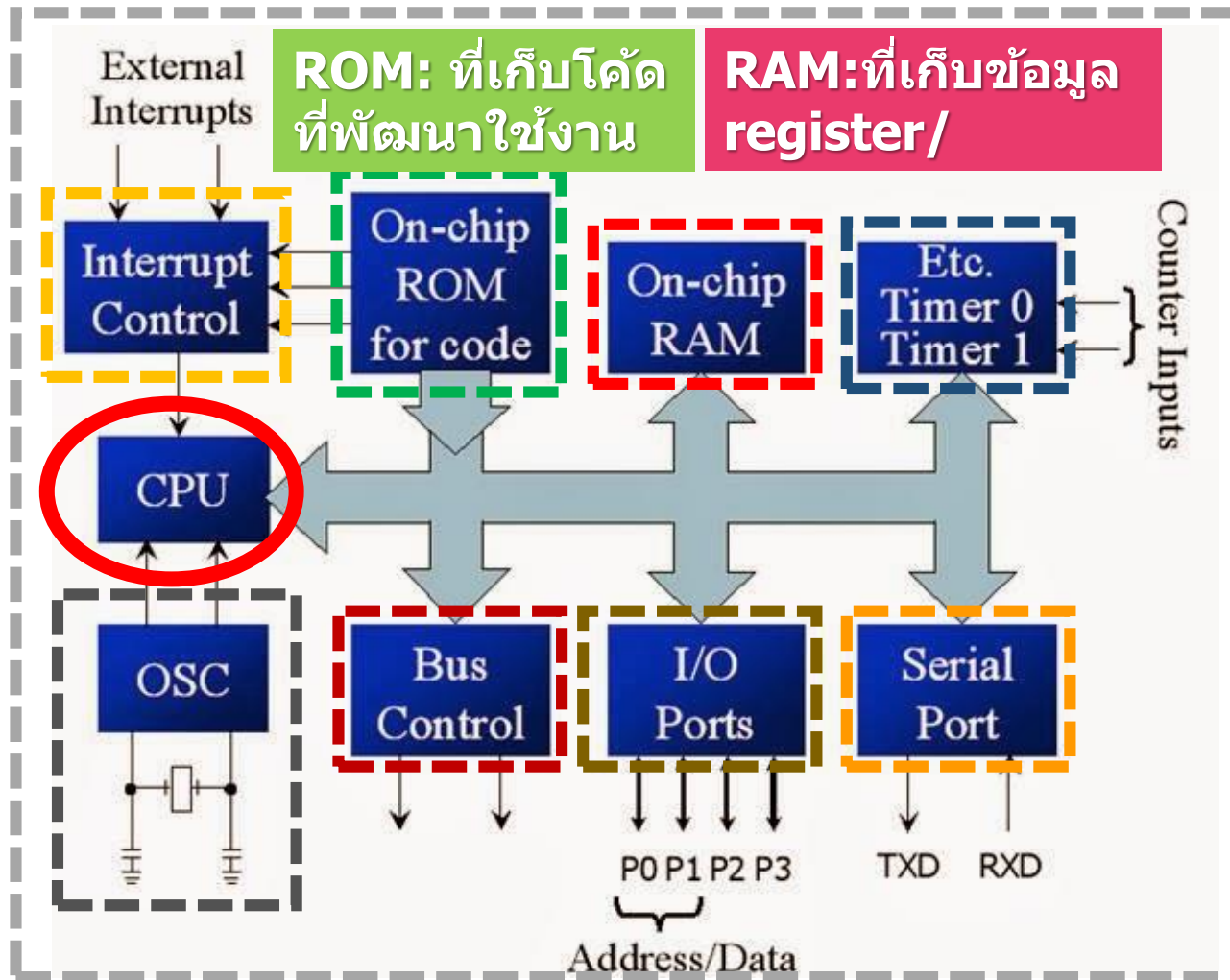
ขั้นที่ 5
recheck
redo



บทที่ 1 สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน

1.6 สถาปัตยกรรมของ MCU ตระกูล MCS-51

MCU กับหน่วยสำคัญต่างๆที่อยู่ภายในชิป (ตัวอย่างชิป: AT89C51)



ส่วนสร้างฐานเวลาเพื่อการนับและการจับเวลา

จุดเชื่อมต่อการสื่อสารอนุกรมกับอุปกรณ์ภายนอก (serial port)

ช่องทางติดต่อกับอุปกรณ์ภายนอก: P0, P1, P2, P3

ส่วนควบคุมบัสข้อมูล บัสตำแหน่ง และบัสควบคุม

ส่วนกำเนิดความถี่ในการทำงาน

ส่วนควบคุมการขัดจังหวะ

ALU + CU



บทที่ 1 สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน

เมื่อจบบทเรียนนี้แล้ว... นักศึกษาควรมีสมรรถนะดังนี้



1.1 มีความเข้าใจ
..บิต-นิบเบิล-
ไบต์-เวิร์ด/
MSB-LSB/
BIN2HEX



1.2 บอกความ
แตกต่างระหว่าง
CPU vs MCU



1.3 อธิบายความ
แตกต่าง
RISC vs CISC
และสามารถระบุ
MCU 4 ตระกูลหลัก



1.4 อธิบายความ
แตกต่างของภาษาที่
ใช้ในการพัฒนา
โปรแกรมควบคุม



1.5 ระบุขั้นตอน
การพัฒนาฮาร์ดแวร์
-ซอฟต์แวร์ บน MCU



1.6 บอกองค์ประกอบหลัก
ในสถาปัตยกรรมของ
MCU ตระกูล MCS-51



มหาวิทยาลัย
ราชภัฏนครปฐม