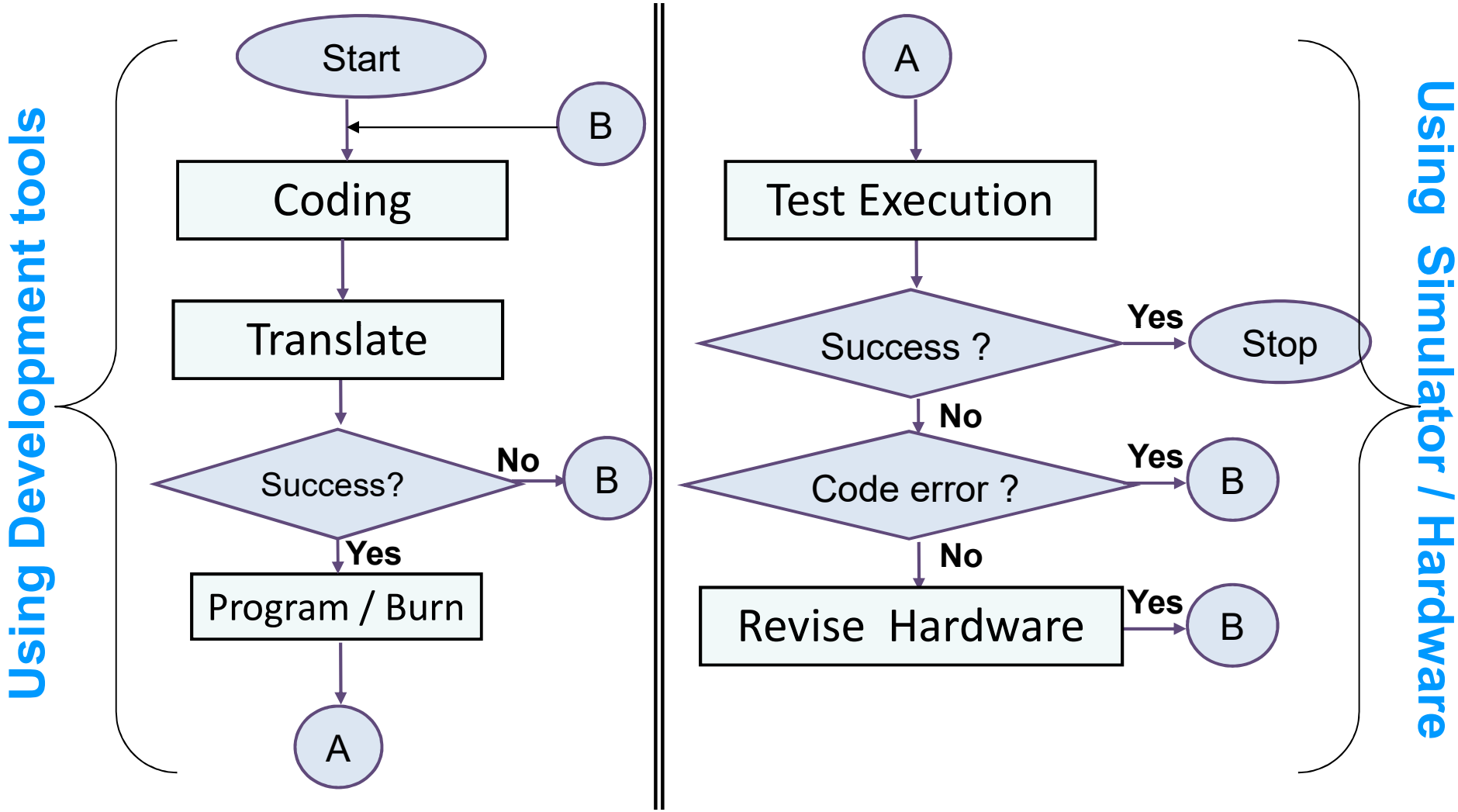
First Program using Keil uVision

WEL Labs, IITB

2016

Programming



Development Tools

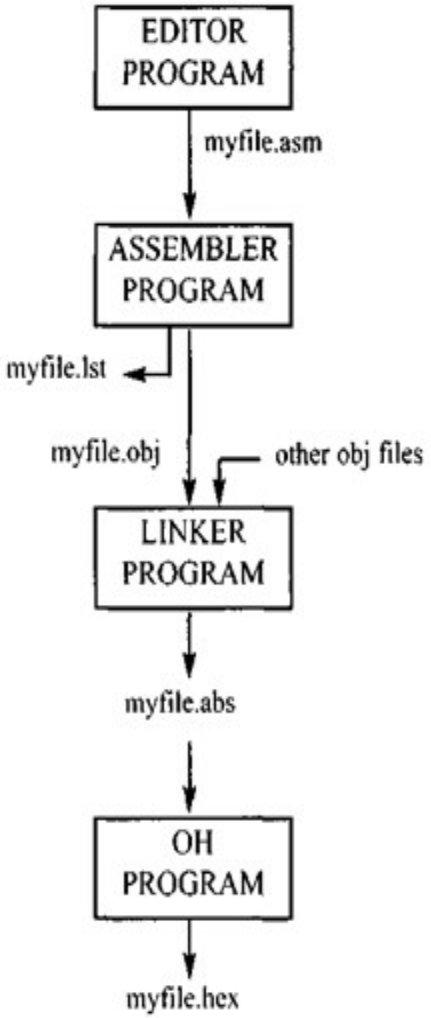
- **Editor –** to enter the program
- **Assembler -** as we are dealing with assembly language initially
- **Execution check –** using Debugger to verify operation of program (on Simulator)
- **Flip Programmer**

  => Put machine code in the chip

Files to be handled

- **.ASM** or **.A51** file (the assembly code)

  LABEL: MNEMONICS ; Comment
- **.OBJ file** holds the machine language code and data for the program can contain relative/unresolved addresses
- **Absolute file (Does not use extension)**
- **.HEX file** holds the machine code in text format suitable for download
- **.LST file** is a line organized file which lists all the codes and addresses as well as errors reported by the assembler



Startup of Program

Setup the program so that the microprocessor finds it when woken up by power turning on.

Locate program in code memory so that control is transferred to it from Reset Vector .

We use assembler directive "ORG" for this

In the Program file :

```
; Setup the Instruction location counter (ILC) of the
; assembler to point to reset vector of  Microcontroller
org 0          ; setup next instruction at address 0
ljmp Main      ; transfer control to our main program
org 100h       ; setup main code at location 100H
Main :         NOP
```

Assembler directive - ORG

Assembler Directive / Pseudo codes:

- Is an instruction to the assembler
- Does not generate binary code but can impact the machine code generated by the assembler

ORIGIN

ORG <<address>>

The ORG directive is used to indicate the the address to use for the next instruction assembly.

- **<<Address>> can be either in hex or in decimal.**
- **If the number is not followed by H, it is decimal and the assembler will convert it to hex.**

Assembler directives - EQU

Label1 EQU Data1

Used to define a constant without occupying a memory location in the data memory space

- **Associates a constant with a label**
- **Does not set aside storage**
- **In Program body all occurrences of the label are substituted by the constant.**

Assembler directives - DB

Label1: DB <<Data List>>

Used to define a 8 bit data (byte) which occupies space in data memory

- **Can associates a label with the data**
- **Multiple data to be separated by commas**
- **Strings terminated with single and double quotes are allowed**
- **Numbers can be in ASCII, Hex or Decimal. Letters (H and D) after the number define the base. Without base numbers are assumed to be decimal.**
- **Numbers should begin with a digit not a character (imp for hex numbers).**

Assembler directives – DB …

```
COUNT:   DB    28    ;Decimal Number 28
N1:      DB    34h   ;Hexadecimal number
STRING1: DB    "ABCD123"  ;ASCII String
```

DW

Label1: DW <<Data List>>

Used to define a 16 bit data (two-byte) which occupies space in data memory

Rules dor <<Data List >> are similar to that of the DB assembler directive

```
N2:   DW    0F34h    ; 16 bit Hex. number
```

Assembler directives - END

END

END

Signals end of the assembler source file to assembler

- **Should be last line of the program**
- **Anything after it is ignored by the assembler**

First Program

TASK

Develop an assembly language program which will cause a LED to blink on one of the Port Pins.

A controllable delay should be achievable before the Led toggles from one state to other

Hands on with First Program …

Listing Files

.LST File

- One Listing file per .asm file is generated by the assembler
- Holds the Program listing in following order
- Location, Object Code, Line Number, Source code
- Listing has Fixed width format and is paginated
- Also Lists any error in the assembly process

.M51 File

- Common listing file is created by linker for the project
- Contains Symbol table and Link Map information

Essential Reading

- **Microcontroller**
  - **Instruction Set**
  - **Supported Memory Map**
  - **Data Formats supported**
  - **Assembler / Compiler capabilities**
- **Peripherals**
  - **Data sheets**
  - **Understand Command**
  - **Understand communication Protocol between it and the MCU.**

Questions ?

Thank you

WEL Labs, IITB

2016

For doubts/errors in this PPT contact :
Suryakant Toraskar e-mail: smtoraskar.iitbombay@gmail.com location : WEL5