Microprocessor based Digital Systems

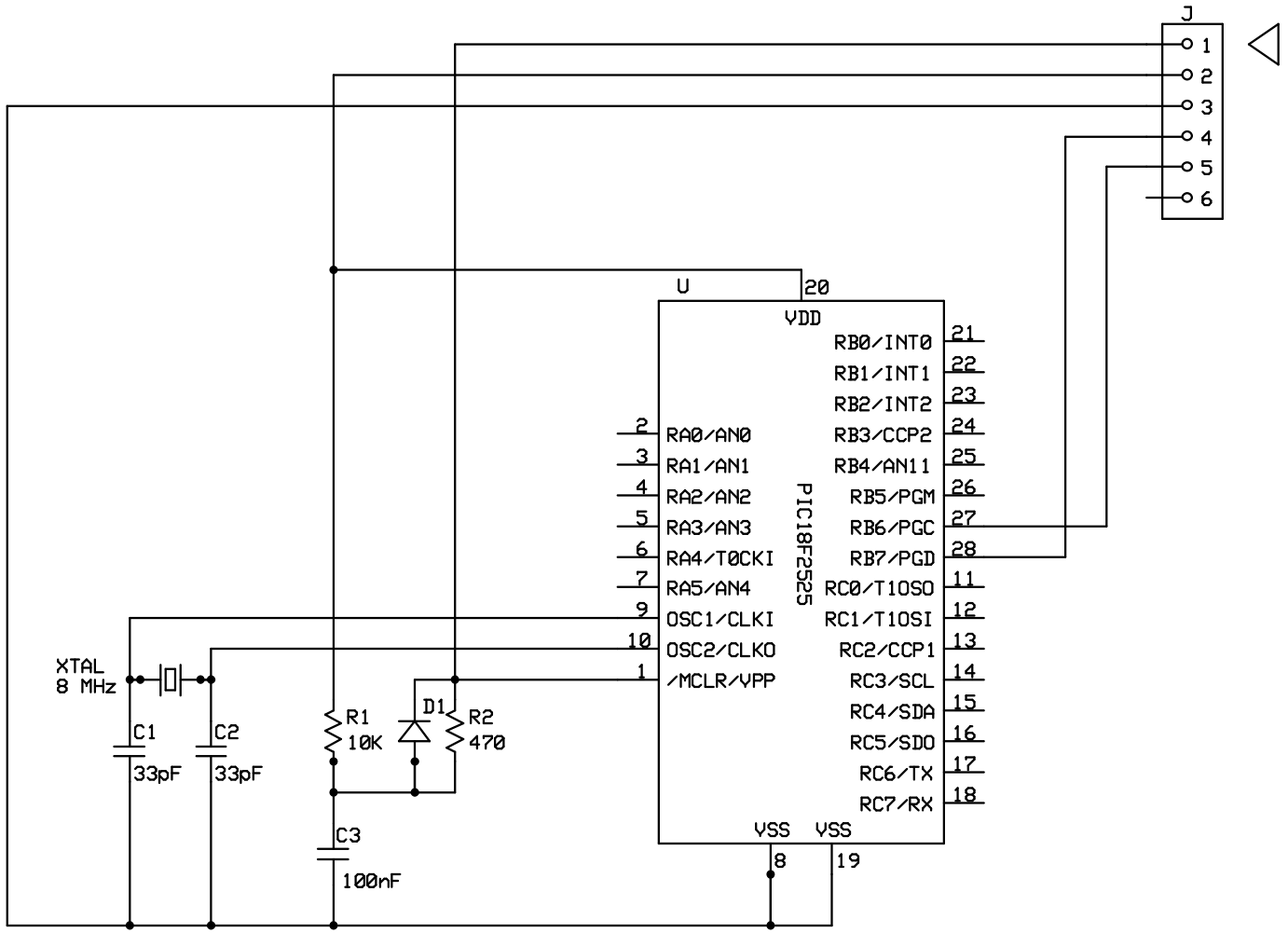
Starters Kit

Guillermo Carpintero

Universidad **Carlos III** de Madrid

Basic Harware

1

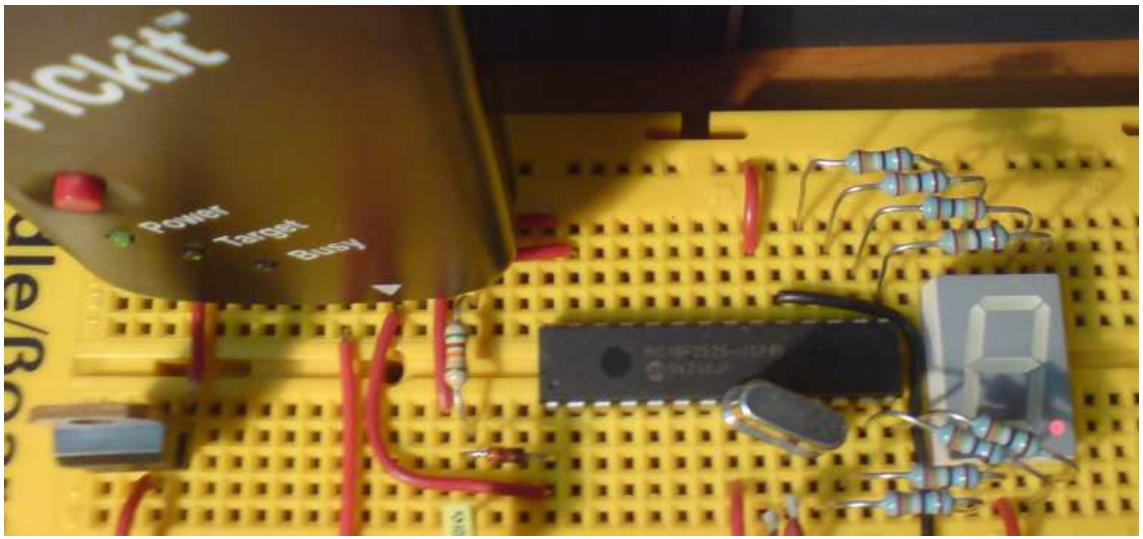


"Vital Elements":

Supply	**VDD**
Ground	**VSS**
Oscillator	**OSC1 y OSC2**
Reset	**/MCLR**

Basic Harware

PICkit 2 Microcontroller Programmer



Integated Development Environment

2

MPLAB

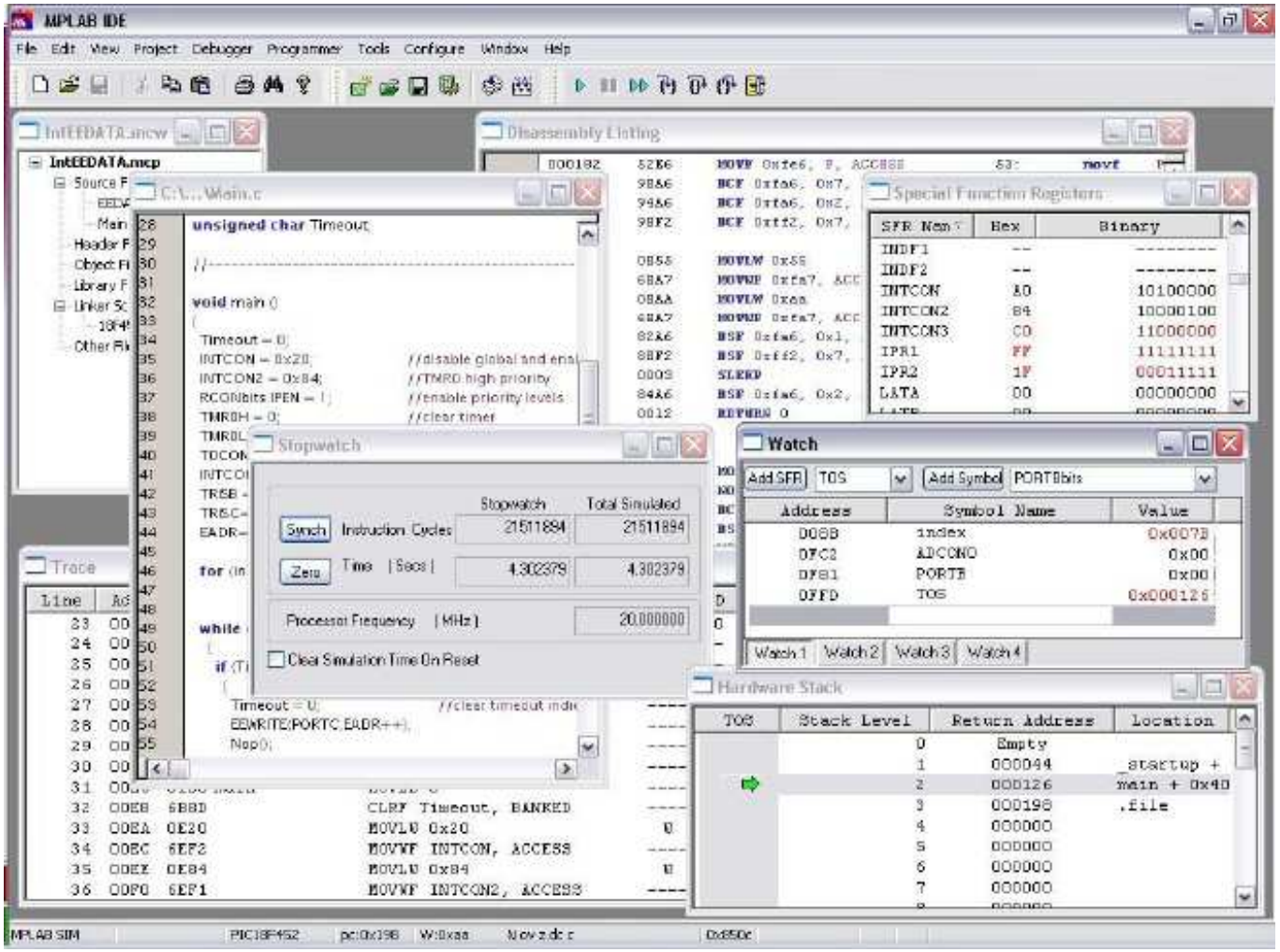
Write

Assemble

Simulate

Program

Debug



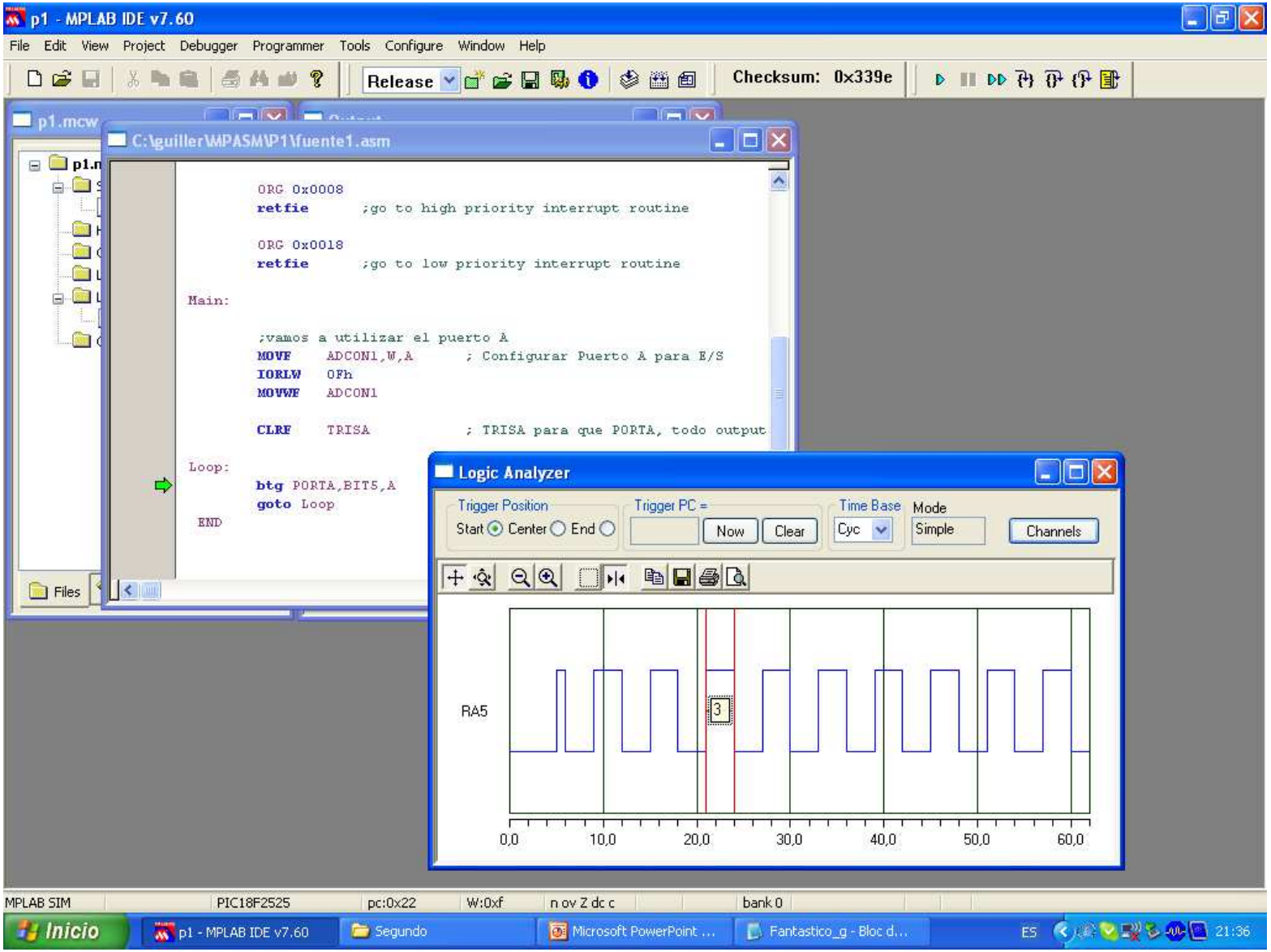
Integated Development Environment

With support (and free of charge if possible)

Introduction to MPLAB IDE	03/30/2004
Tips and Tricks Using MPLAB v6.61	09/16/2004
Introduction to Microchip's Development Tools	02/17/2004
Choosing a Debug Tool	02/24/2006
Introduction to the MPLAB Visual Device Initializer (VDI)	08/26/2004

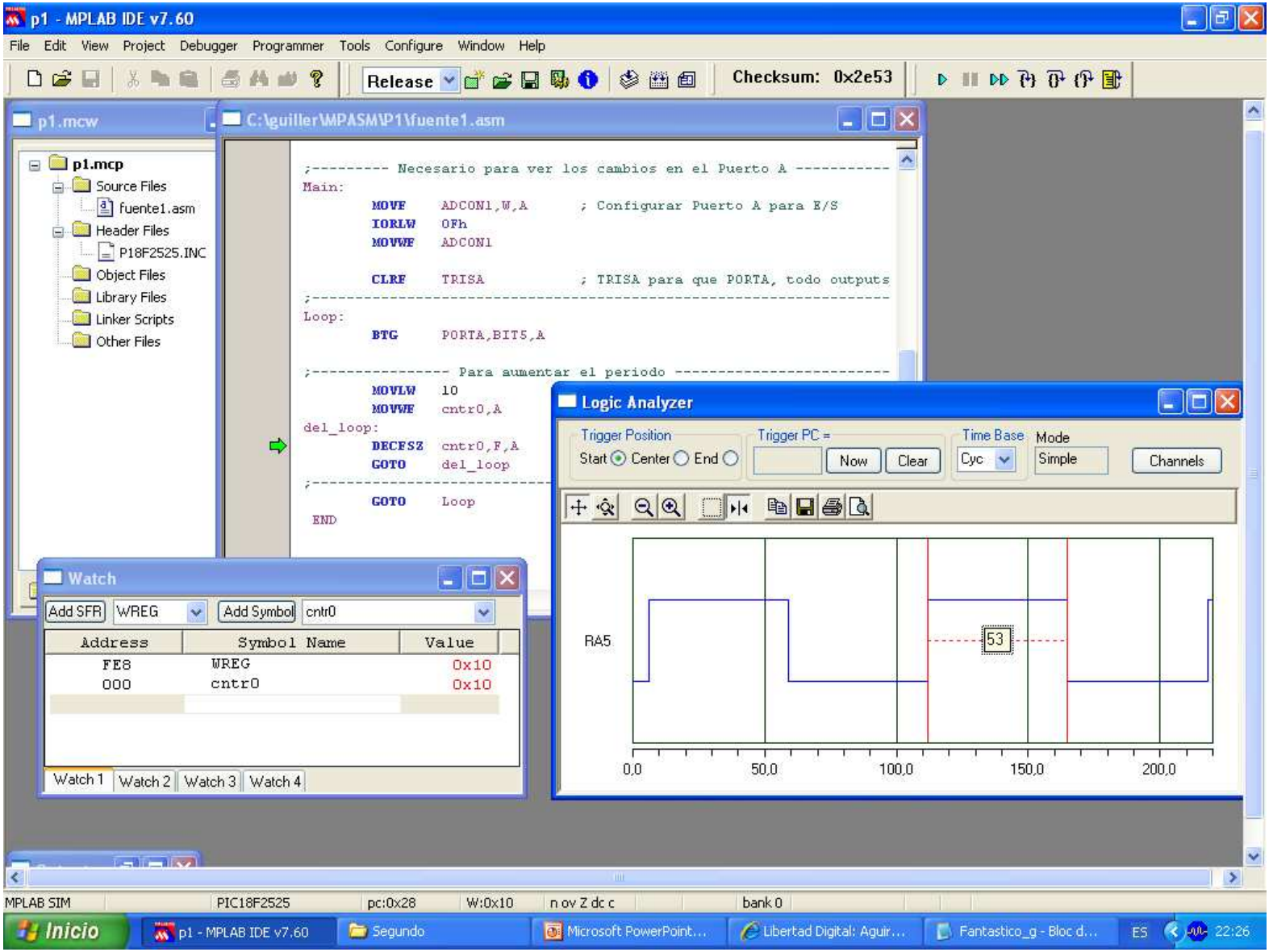
Integated Development Environment

Example (I)



Integated Development Environment

Example (II)



Templates

3

Avoid the the challenge of a *blank sheet* ! ! !

```
;                                                                               *
;    Filename:                                                                  *
;    Date:                                                                      *
;    File Version:                                                              *
;                                                                               *
;    Author:                                                                    *
;    Company:                                                                   *
;                                                                               *
;********************************************************************************
;                                                                               *
;    Files required:                                                            *
;                                                                               *
;********************************************************************************
        LIST P=18F2525                  ;directive to define processor
        #include <P18F2525.INC>;        ;processor specific variable definitions

;********************************************************************************
;Configuration bits
;Microchip has changed the format for defining the configuration bits, please
;see the .inc file for futher details on notation.  Below are a few examples.

;   Oscillator Selection:
    CONFIG  OSC = HS            ; alta velocidad del PIC
    CONFIG  WDT = OFF

;********************************************************************************
;Etiquetas
label           equ     expr


;********************************************************************************
;Reset vector
; This code will start executing when a reset occurs.

                ORG     0x0000
                goto    Main                            ;go to start of main code

                ORG     0x0008
                retfie                                  ;go to high priority interrupt routine

                ORG     0x0018
                retfie                                  ;go to low priority interrupt routine
```

Important!
System configuration

```
Main:
                ;Inicializacion en el PIC para utilizar los pines del puerto A como E/S
                MOVF    ADCON1,W,A
                IORLW   0Fh
                MOVWF   ADCON1

                CLRF    TRISA                   ; TRISA para que PORTA, todo outputs
```

```
PPAL:
                                                Infinite Loop
                goto PPAL

 END
```

Flux Diagrams

4

Help to organize the program before writting it

Allow to organize the program flow (sequence of instructions).

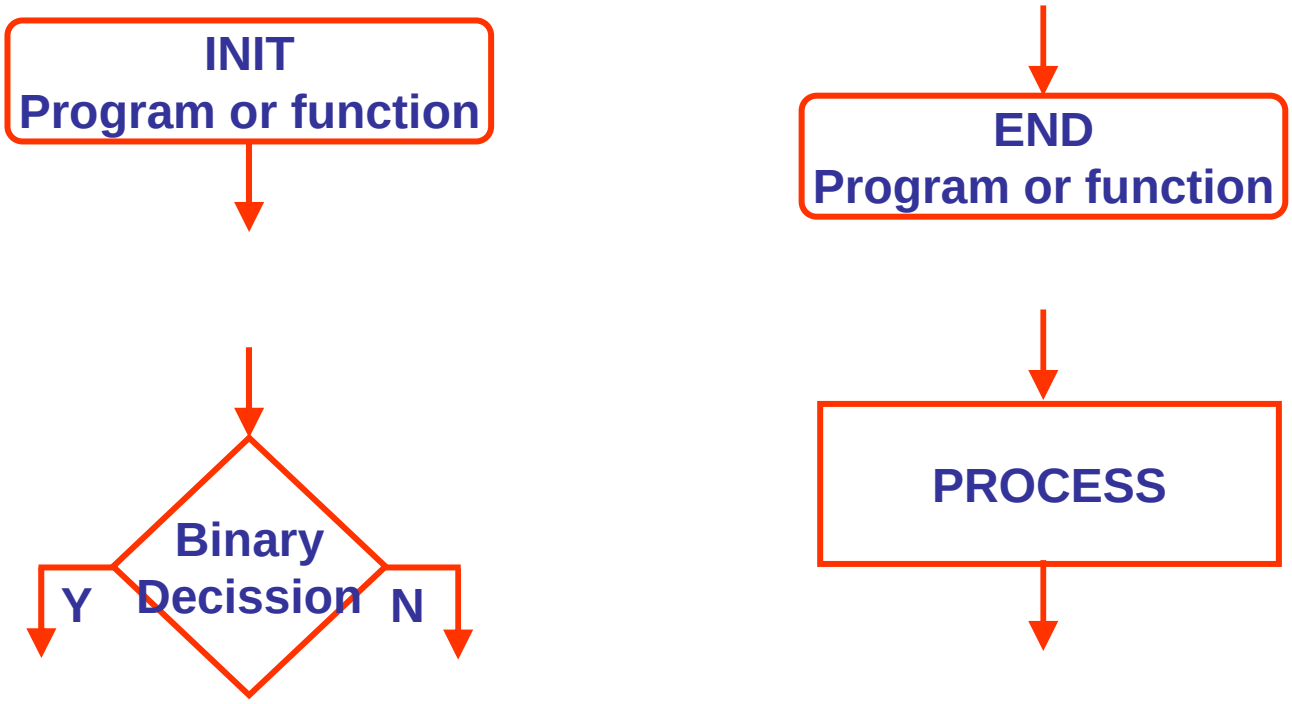
It is a tool to:

design
document a program

Use standard symbols that are connected between arrows to indicate the couse of the program execution (sequential exacution principle) of the program instructions.

Flux Diagrams

Symbols



Programming Techniques

5

¿Why use programming techniques?

It is very dificult to write **good** code.

¿What defines a **good** code?**:**

Program memory usage
Execution time
Readable, and reusable
Delivered on time

Techniques:

~~Trial and Error~~

Structured Programming

Programming Techniques

Structured Programming

Flux diagram allow an infinite ways of being connected.

Structured programming is a discipline which basically limits the possible flux diagram combinations that we can use.

Allowed structures:

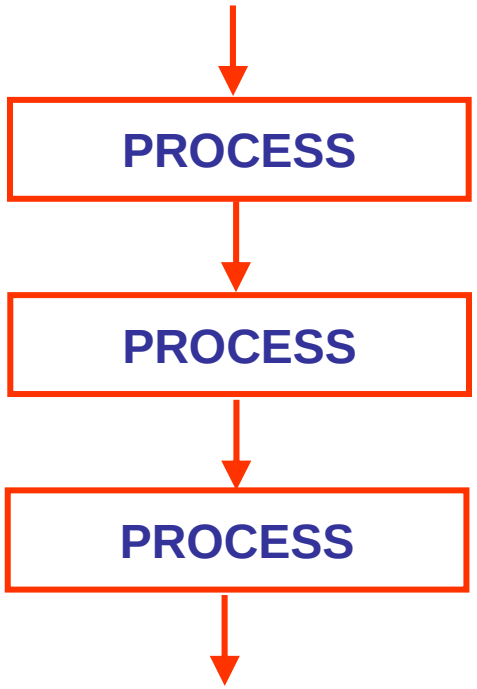
SEQUENCE

IF-THEN-ELSE

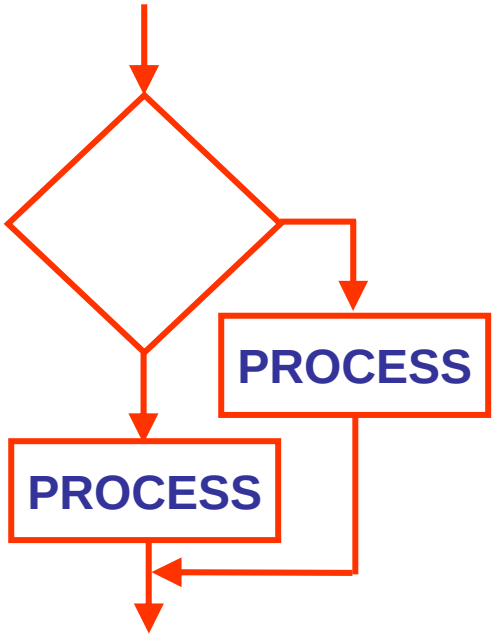
DO-WHILE

Programming Techniques

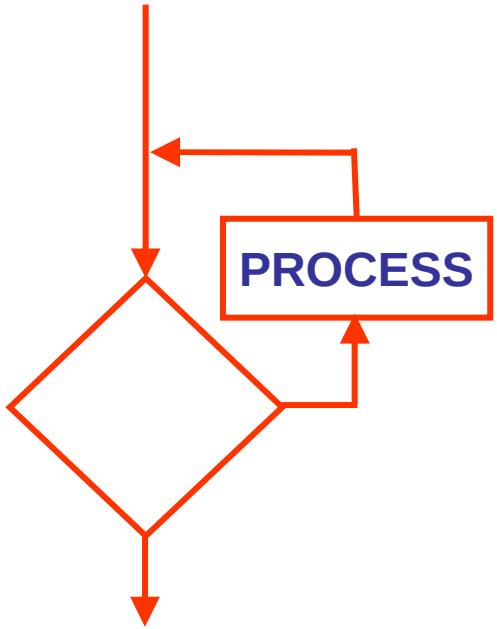
SEQUENCE



IF-THEN-ELSE

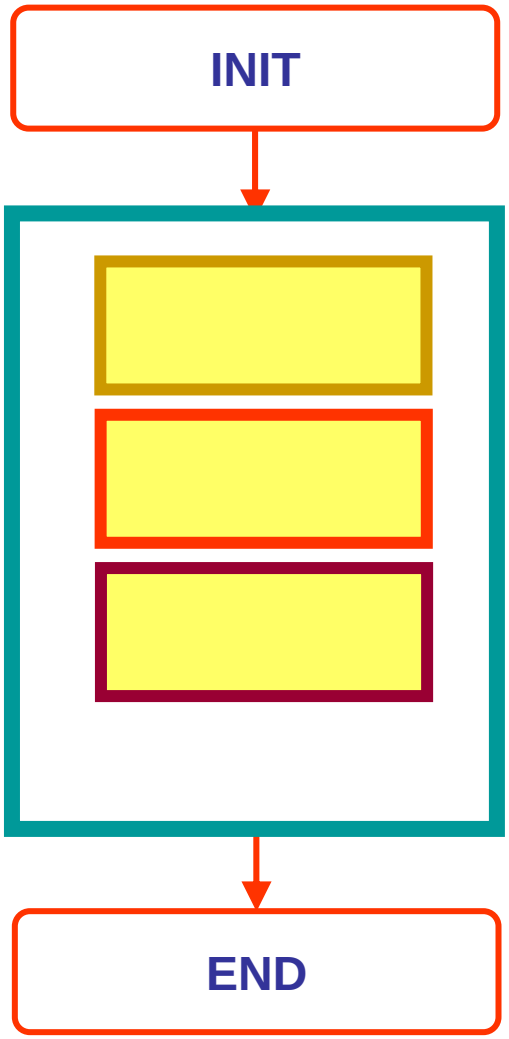


DO-WHILE



What do all these structures have in common?

Top-Down Design



Divide and Conquer

Contruct your program as a sequence of **functions**