Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

EEU33C08 Digital Electronic Design Project

Specify -> Partition -> Design -> Capture - > **Verify** -> Build -> **Test** ->
Debug -> Document -> Present -> Use & Maintenance -> **Dispose**

Eugene O'Rourke

Date 2023-JAN-16

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Multivibrator Circuit Class – Astable, Monostable, Bistable

Monostable Description

Astable multivibrator: in which the circuit is not stable in either state — it continually switches from one state to the other, e.g. clock circuit.

Monostable multivibrator: in which **one** of the states is stable, but the other state is unstable (transient). A trigger pulse causes the circuit to enter the unstable state. After entering the unstable state, the circuit will return to the stable state after a set time. Such a circuit is useful for creating a timing period of fixed duration in response to some external event. This circuit is also known as a **one shot**.

Bistable multivibrator: in which the circuit is stable in either state. It can be flipped from one state to the other by an external trigger pulse. This circuit is also known as a flip-flop. It can store one bit of information, and is widely used in digital logic and memory.

Astable multivibrator(1/1)

- Electronic equivalent of a pendulum/metronome
- Typically used to implement clock circuits
- Want all latches/flip flops in a circuit to update or change at same time(synchronous)

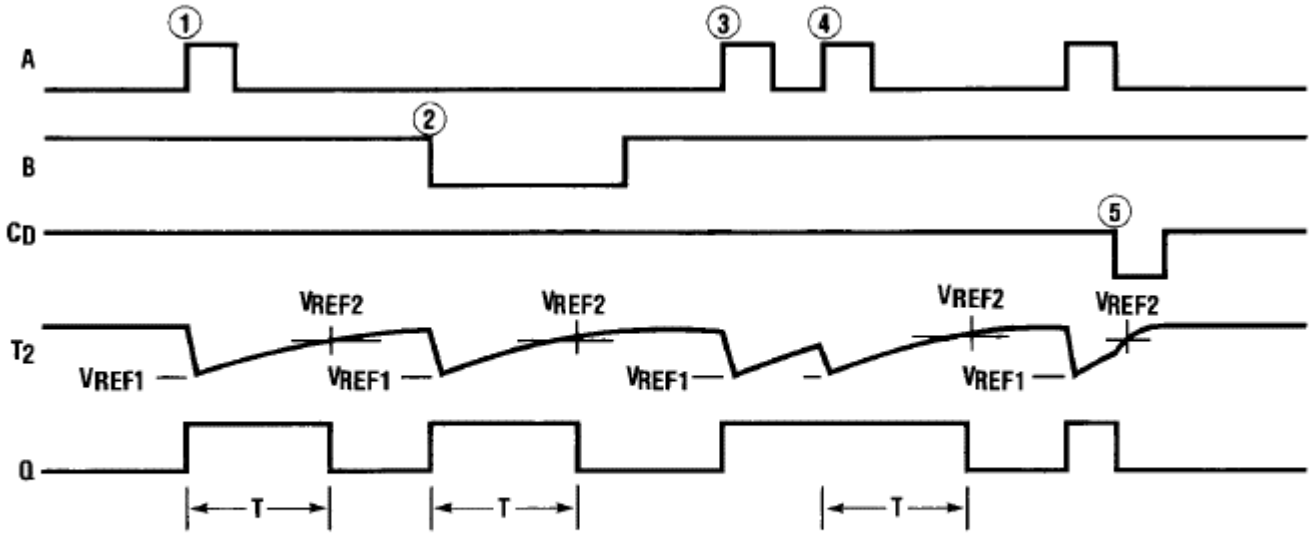
(a) Response to positive level

(b) Positive-edge response

(c) Negative-edge response

Monostable multivibrator(1/3)

- CD4538 is a dual, precision monostable multivibrator with independent trigger and reset controls.
- Generates a pulse whose duration can be set in response to an external pulse

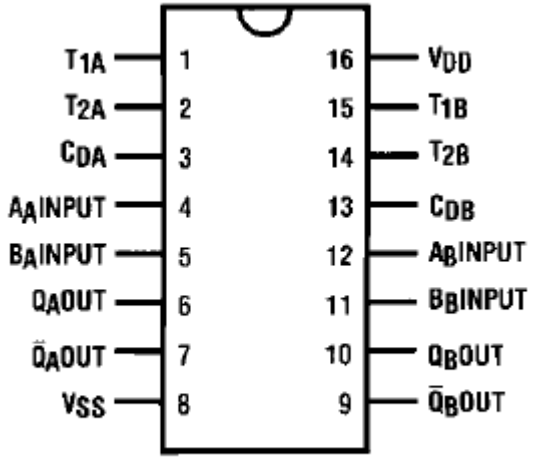


Monostable multivibrator(2/3)

Inputs			Outputs	
Clear	A	B	Q	$\overline{Q}$
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	↓	⎍	⎍
H	↑	H	⎍	⎍

H = HIGH Level
L = LOW Level
↑ = Transition from LOW-to-HIGH
↓ = Transition from HIGH-to-LOW
⎍ = One HIGH Level Pulse
⎍ = One LOW Level Pulse
X = Irrelevant

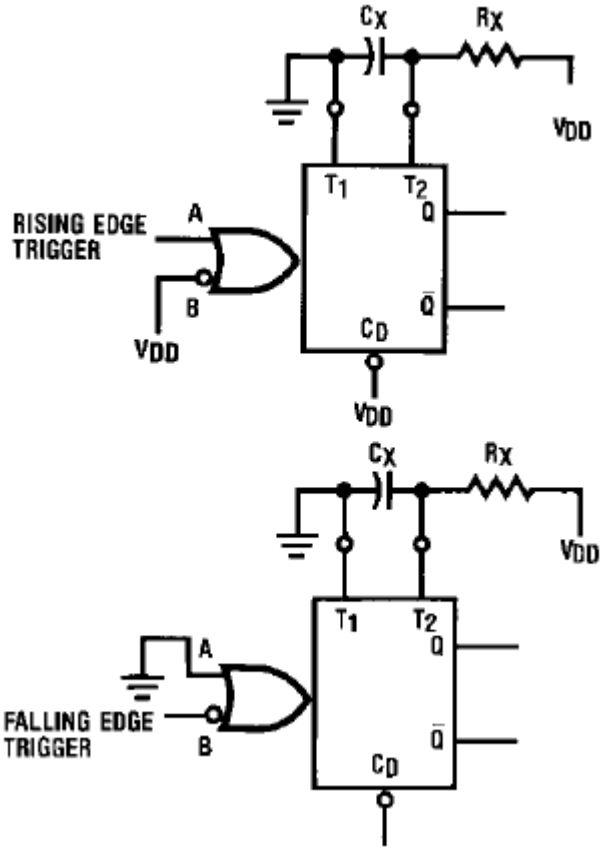
Pin Assignments for DIP and SOIC



Top View

Monostable multivibrator(3/3)

- Output Pulse Width is a function of the product of R & C



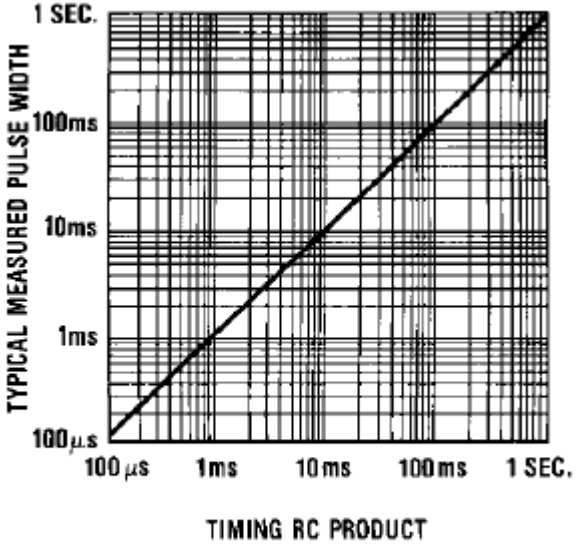
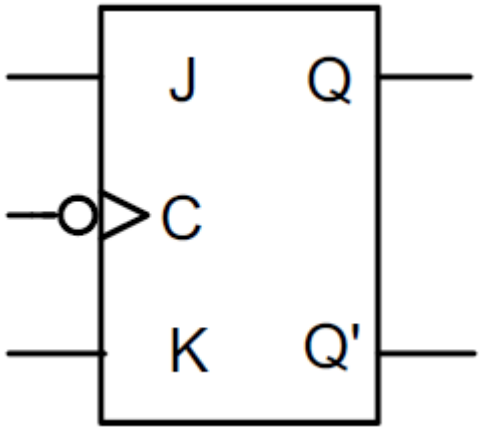


FIGURE 11. Typical Pulse Width Versus Timing RC Product

Bistable multivibrator/JK Flip Flop(1/2)

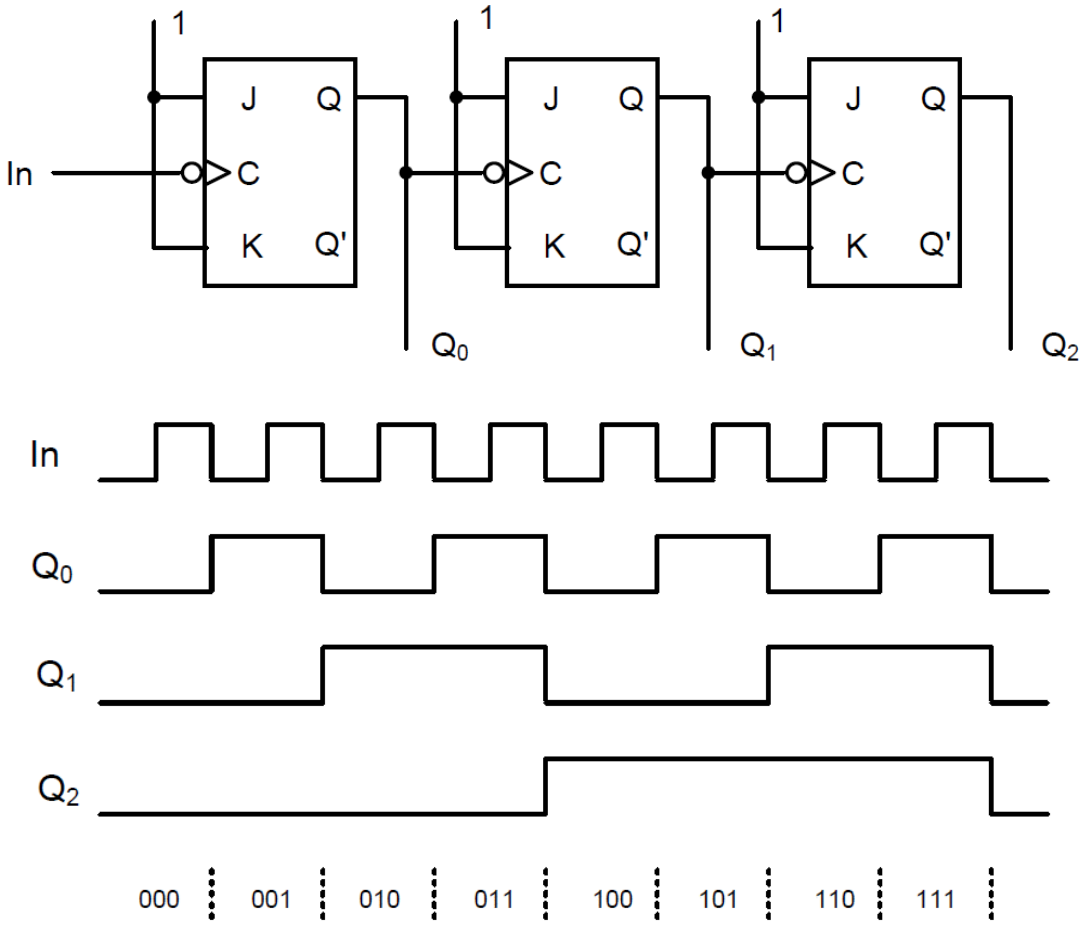
- Example is 4027
- Inputs are **J**, **K** & **C**lock
- Outputs are typically **Q** and $\overline{\mathbf{Q}}$ (pronounced Q-bar, also ~Q or Qn)

J	*K*	*Q(N+1)*	**Comment**
0	0	Q(N)	No change
0	1	0	f-f resets
1	0	1	f-f sets
1	1	Q′(N)	f-f reverses state



- Note the “bubble” on the clock

Bistable multivibrator/JK Flip Flop(2/2)

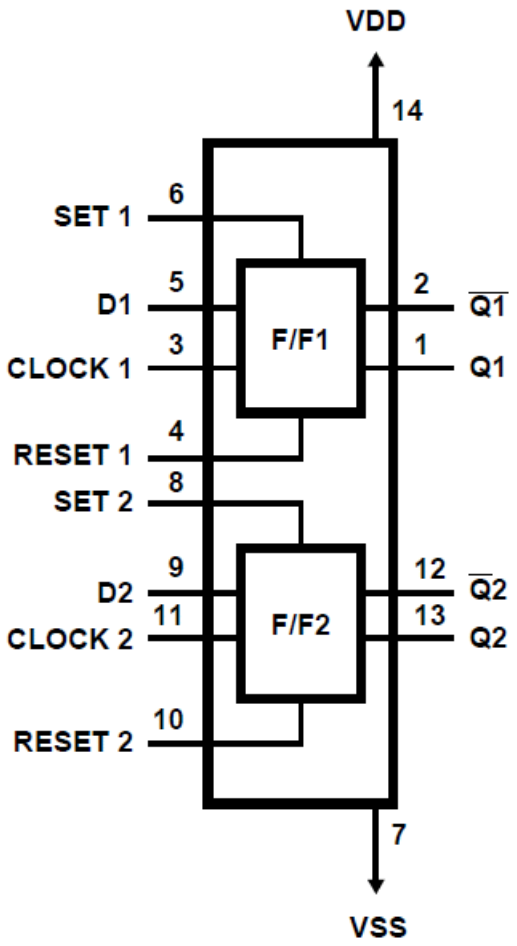


Bistable multivibrator/D-Type Flip Flop(1/2)

- We will present ***simplified*** d-type flip flop functionality as it has been derived & covered in greater detail in 3C7
- Implements ***volatile*** "triggered" storage vs "addressed" 1-bit storage
- Inputs are **D**ata, **C**lk, **R**eset(**S**et), **En**able
- Inputs can be active-low or active-high.
- Outputs are typically **Q** and $\overline{\mathbf{Q}}$ (pronounced Q-bar, also ~Q or Qn)
- Reset sets **Q** to 0 Volts or logic '0', $\overline{\mathbf{Q}}$ to VDD or logic '1'
- If certain timing constraints are met, the logic level(data bit) presented at the D input gets passed to the Q output at a rising or falling Clk **edge** after a finite time. We say this is 'edge-triggered'
- Other inputs are 'level-sensitive' i.e. respond to 1/0[VDD or 0 volts]

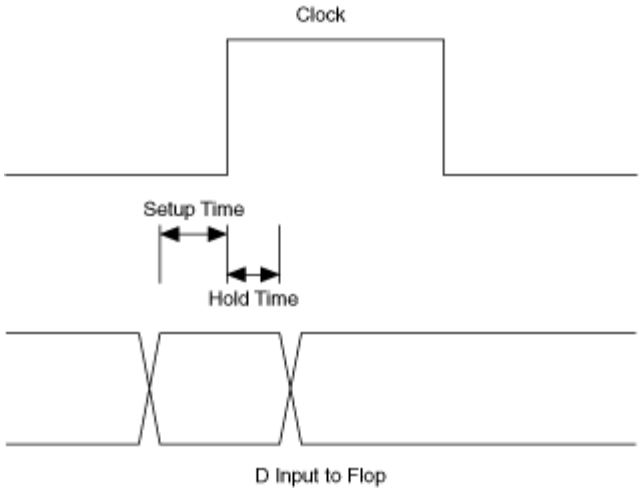
Bistable multivibrator/D-Type Flip Flop(2/3)

- 4013 consists of two identical, independent data type flip-flops
- Each flip-flop has independent data, set, reset, and clock inputs and Q and Q outputs
- **These devices can be used for shift register applications, and, by connecting Q output to the data input, for counter and toggle applications**
- The logic level present at the D input is transferred to the Q output during the positive going transition of the clock pulse
- Setting or resetting is independent (**asynchronous**)* of the clock and is accomplished by a high level on the set or reset line, respectively
- * It is possible to have synchronous set/reset



Bistable multivibrator/D-Type Flip Flop(3/3)

- **Timing Constraints to ensure correct operation**
  - Setup time
    - Minimum time for which D input must be maintained at constant value prior to transition of the clock
  - Hold time
    - Minimum time for which the D input must not change after the clock transition
  - Clock-to-q delay
    - time to propagate value of D at rising/falling edge of clock
    - It is not anticipated these will cause you any challenge in this project.



Project 1 – Combination Lock

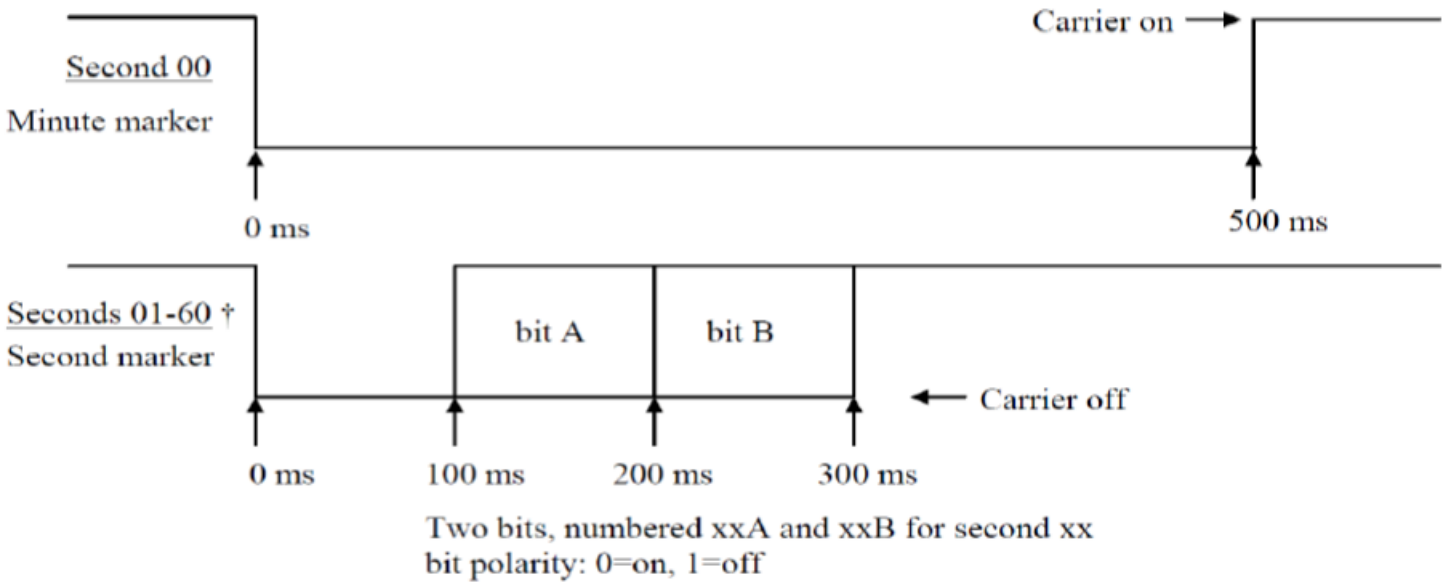
- 4 Digit Combination Lock
- Code entered using a numeric keypad
- When the correct code is entered the lock should unlock for 5 seconds
- When the incorrect code is entered and alarm should sound for 5 seconds
- When the incorrect code is entered 3 times the lock should be disable for a number of minutes
- Sequence matters

Project 1 – Challenges

- Maintaining a history entered digit values
- Logic to check the entered code against the unlock code
- Logic to unlock and sound the alarm
- Logic to disable the lock after 3 incorrect entries
- Interfacing with the Numeric Keypad

Project 2 – Rugby Signal

- Receive and decode the Rugby clock signal.



Project 2 – Challenges

- Detect the beginning of the signal frames
- Decode the bits embedded within the frame and extract the time information
- Display the time & not date

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Project Management

Structured Time Management

- People leave, opt-out, get sick, die, the weather, software failure, component lead time
- Build contingencies into your plan so that despite issues you pre-empt failure and **still** finish on time.
- It's always later than you think
- New Structured Time Management
  - Start with Deadline and work backwards
  - Generate Milestones for yourselves
- The key to good project management is the timely discovery of the unknown unknowns(Quinn 2006)
- Experience always comes too late for when you need it.(Quinn 2006)
- Nothing is impossible, the impossible just takes longer and costs more(Quinn 2006)

Structured Time Management

Week 13 Report Submission

Week 11 Demonstrations

Week 10 You should be finished

Week 9 You should be debugging top level

Week 8 You should be completing the build

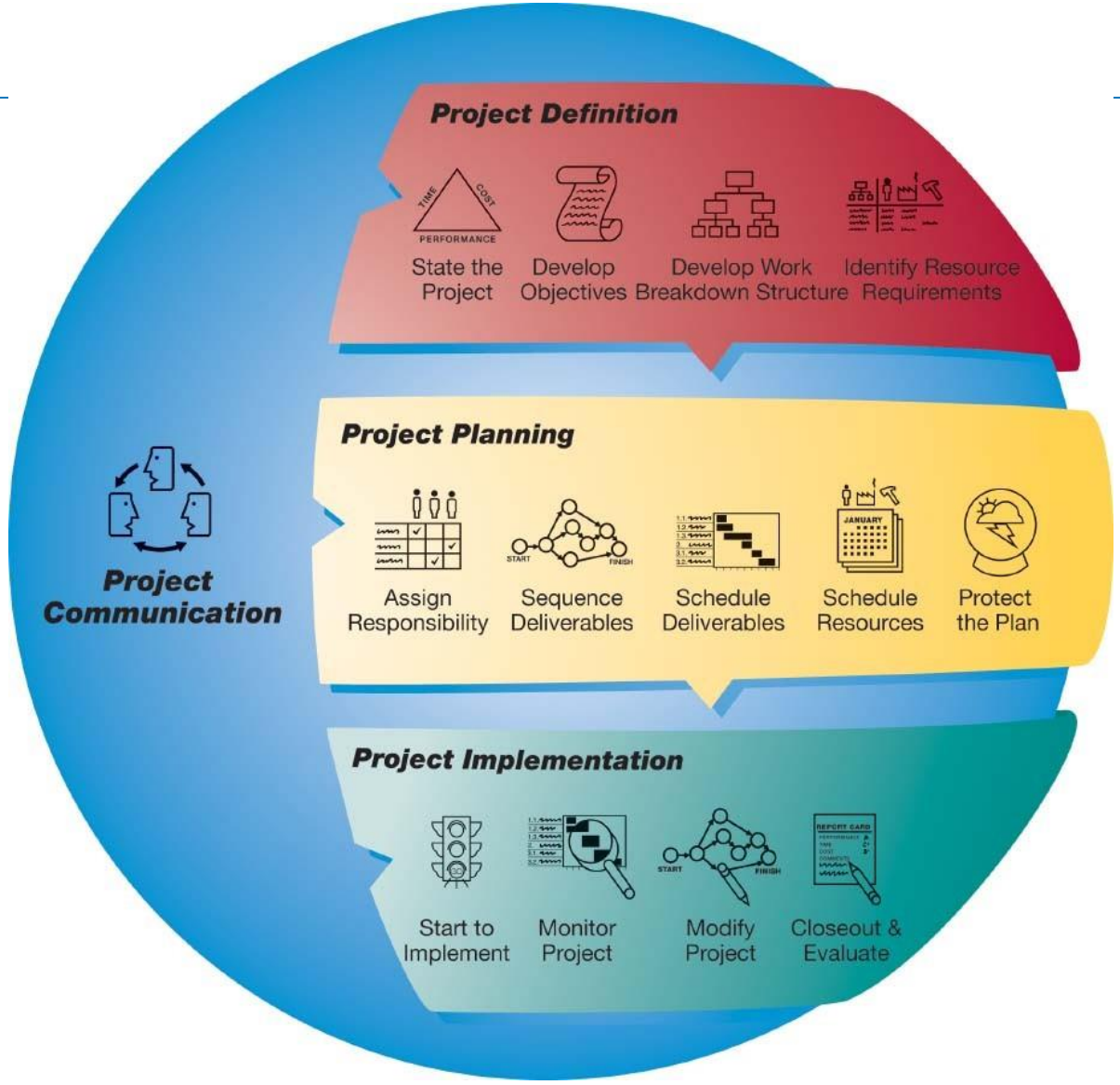
Week 6 Sub-block build debug

Week 5 Sub-block build complete

Week 4 Subblock Multisim design **+ Submit Arduino test plan**

Week 3 Partition design sensibly **+ Begin Arduino test plan**

Structured Project Planning(Source: Kepnor Tregoe)



Structured Design – Fletcher Methodology

- Not published anywhere
- William I. Fletcher “An Engineering Approach to Digital Design”
- Facilitates logic partitioning, resource allocation, planning, documentation, reviews, implementation, reusability, verification and **signoff** (Is your work worth $1,000,000?)
- Design methodology of choice for Digital Equipment Corporation (DEC) and all spinoffs, e.g. Parthus, Crevinn, Microsemi, 3Com, Black Box.
- Projects start with an intensive, collaborative, recursive discussion phase amongst all engineers to agree on sub-blocks and signals
- **Nothing** is implemented until this phase is complete
- Confidence and clarity should increase as the deadline[SIGNOFF] approaches!

Synchronous vs Asynchronous

- Here we focus on the relationship between the clock and the data
- Synchronous signals have a known fixed relationship with system clock
- Asynchronous signals are independent of the clock and can happen at any time.

System Analysis

- Here we focus on the logic view of a system.
- What the system is to accomplish.
- To help describe the behavior and to allow us to functionally decompose a design, we use data flow diagrams.
- Your first job is to meet the specification
  - Do not get sucked into “feature creep”

System Analysis

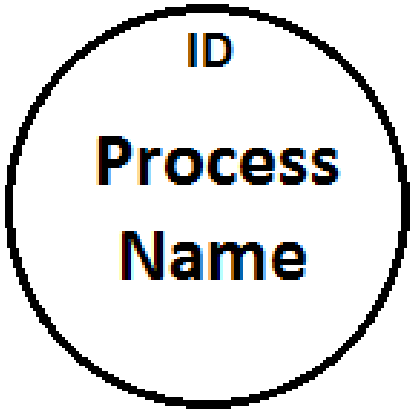
- Result is design breakdown between sub-blocks with defined interfaces
  - Inputs
  - Outputs
  - Bidirectionals
  - Active high or Active low?
- Starting point for either design or repeat the process to sub-divide further
- Starting point for building Arduino test program

Data Flow Diagrams (DFD)

- Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs
- Focus is on the movement of data between external entities (anything outside of your design) and your processes and between processes and data stores

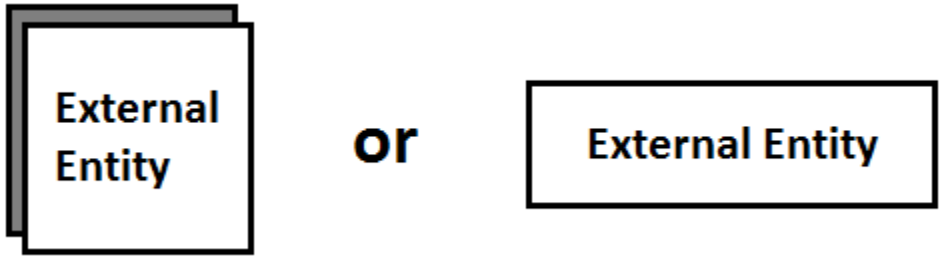
DFD Notation

- Process
  - A process transforms incoming data flows into outgoing data flows
  - Ensure it has suitable name describing what it does.



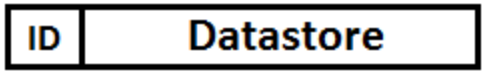
DFD Notation

- External Entity
  - External entities are objects outside of the system/design. They are sources and destinations of the systems data flow inputs and outputs



DFD Notation

- Datastores
  - Datastores are repositories of data in a system (i.e. RAM or a file)
  - Ensure it has a suitable name describing it



DFD Notation

- Data flow
- Data flows are connections through which data flows
  - Ensure it has a suitable name that describes its function. These will become signal names in your circuit or HDL(Verilog or VHDL).
- There are two types
  - Single bit data which is represented by a dashed line
  - A multibit bus which is represented by a continuous line

DFD Layers

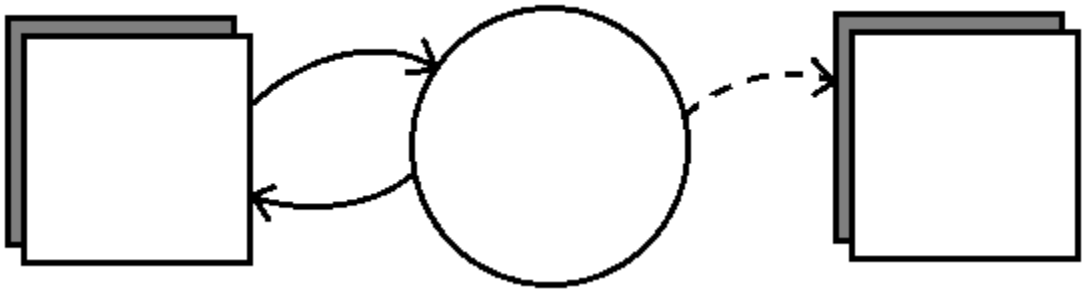
- Data flow diagrams are nested in several layers
  - Starts with a single process on a high level diagram. This is called a CONTEXT DIAGRAM.
  - Other lower layers expand to show more and more detail.

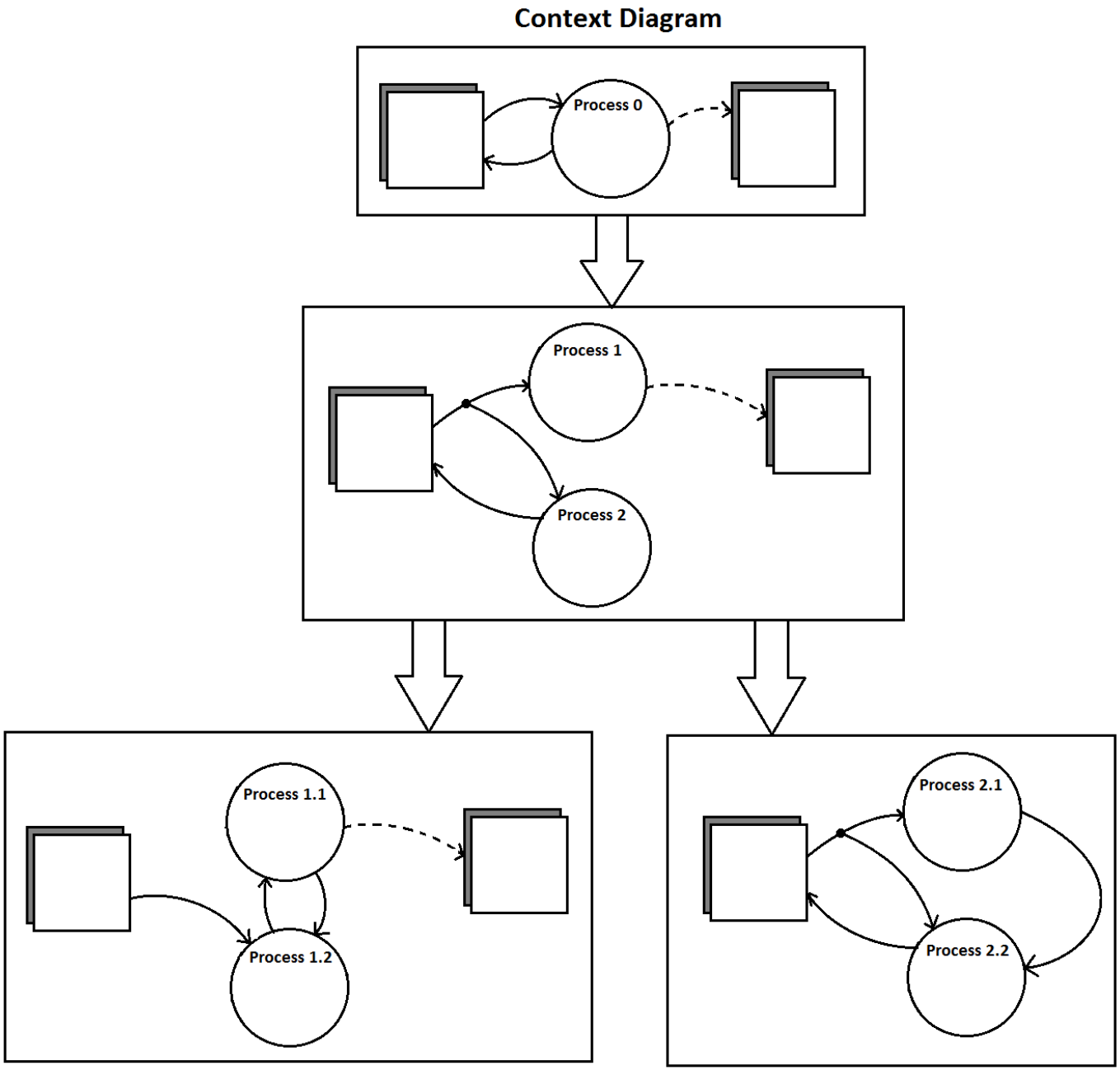
Context Diagram

This defines the scope of the system by identifying the system boundary

Contains

- One process (which represents the whole system)
- All external entities
- Shows data flows linking the process to the external entities




Context Diagram
Process 0
Process 1
Process 2
Process 1.1
Process 1.2
Process 2.1
Process 2.2

Data Dictionary

- A data dictionary is used to identify each new data flow and datastore in a DFD
- Document block interfaces
- Signal Naming
  - Active High = Reset
  - Active Low = Reset_n

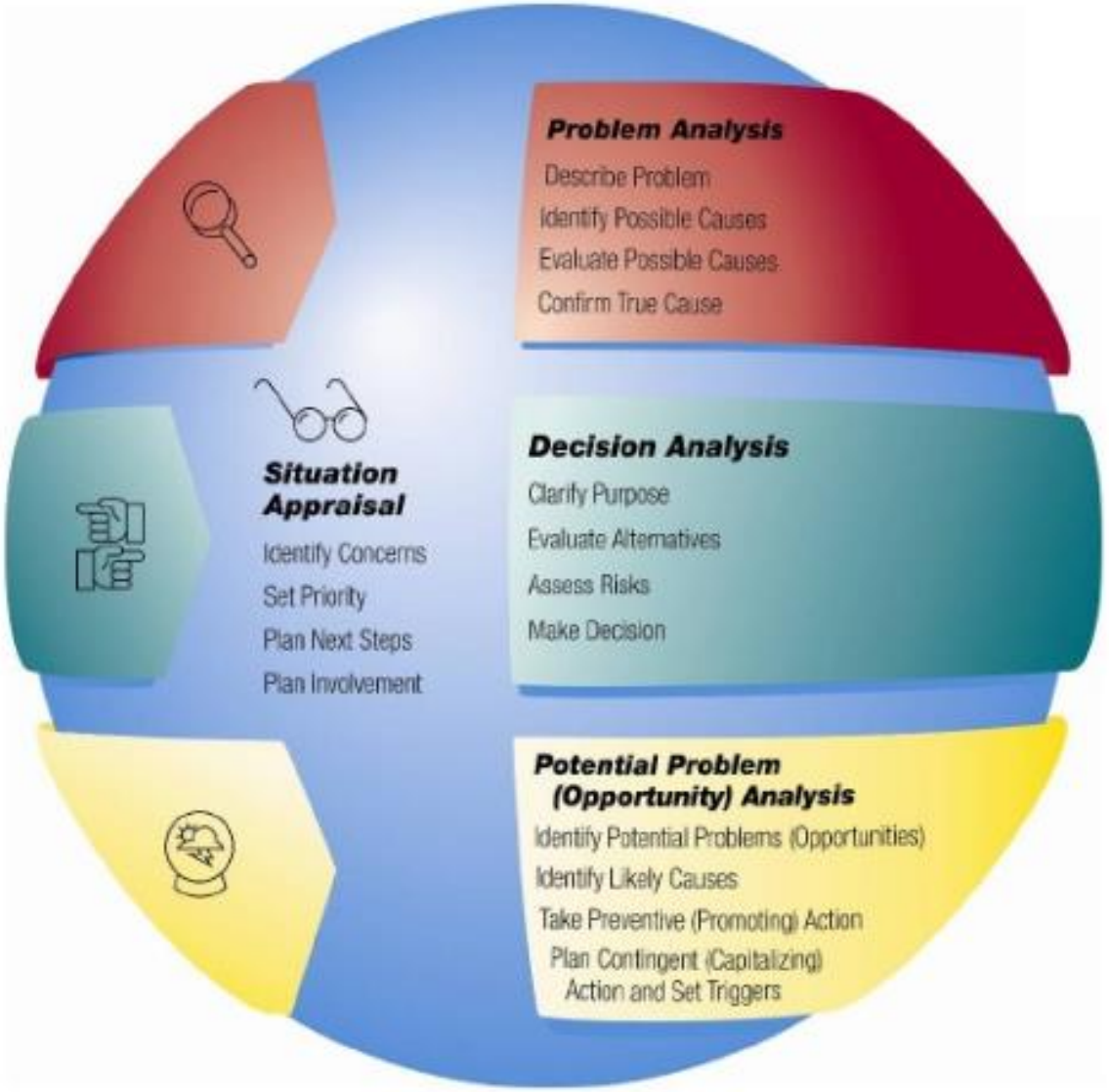
reset (H) : When asserted , clears all counters to 0

Indicates the signal is active high

Structured Unit Testing

- Read Testing Primer on Blackboard
- Simulate chosen 4000 series parts in Multisim on their own and reconcile operation with datasheet.
- Build Arduino test programs to stimulate each sub section and check it's outputs
- Rotate design and testing between group members
- Be mindful of “edge” cases, e.g. rollover, underflow, overflow etc

Structured Problem Solving(Source: Kepnor Tregoe)



Structured Problem Solving

- Trust But Verify
- You should not rely on serendipity to make sure your design works!
- Confidence & clarity should increase as you get closer to the deadline
- Define the problem
  - What it is
  - What it isn't
  - Build positive and negative tests for all of these
  - Record the results
- Minimise components => less wiring => less to debug, not less debug
- Every IC adds 16 wires which have to be tested for each issue, working or not working
  - 8 chips = 128 wires
  - 1 minute per wire = **2 hours** per issue!
  - Colour code/label your wiring!

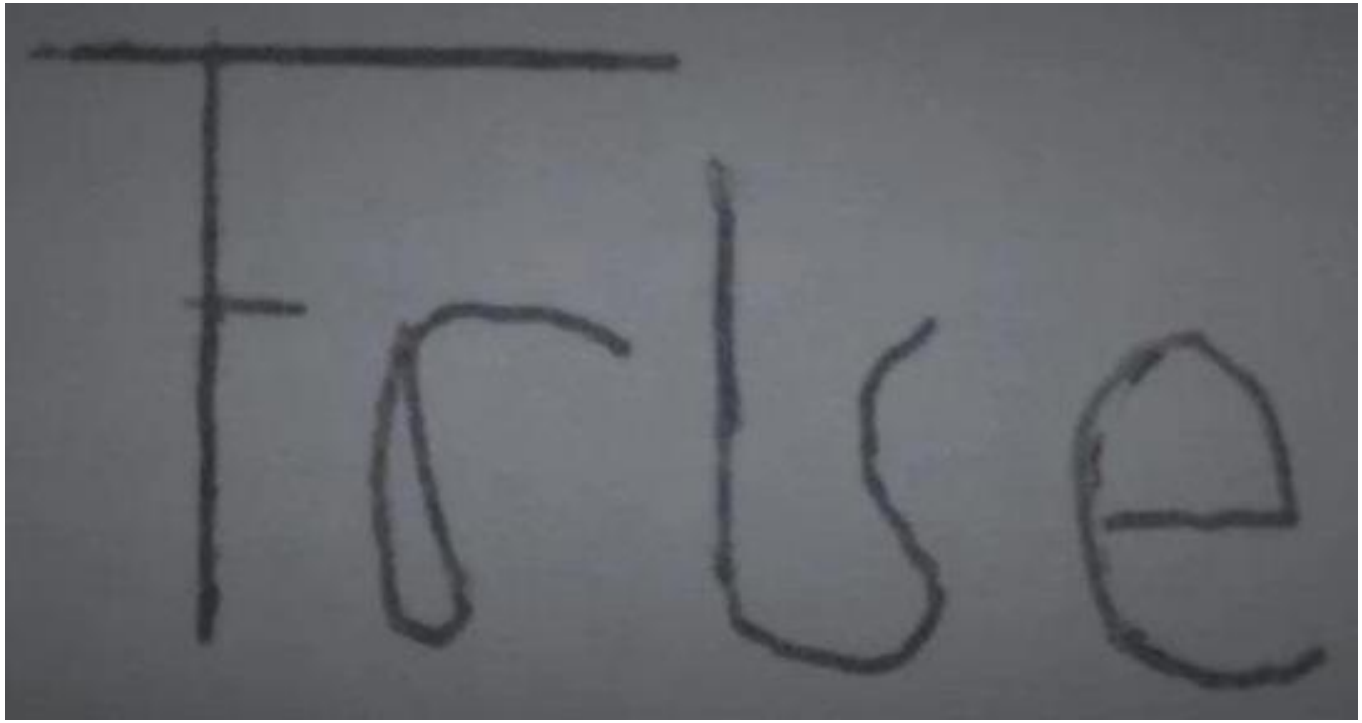
Structured Revision Control

Structured Revision Control - Meaningful & Consistent file naming

- yyyy_mmm_dd_[ID number|group number]_p0p001
- Do not use:
  - _bak
  - _Thursday
  - _Thursday_bak etc
- Remove unwanted trial files as soon as you have a fixed solution.

Structured Communication

- Be Clear!



- If I can't read it, I can't mark it.

Structured Communication

- Communication for Action
- Meet regularly to ensure everyone understands what is going on.
- ***"Craft"*** emails to get the response you want
- Meaningful, Sensible, Email communication
  - Subject: 3C8
  - Sub-Subject: S/W or H/W or Attendance etc
  - Example
    - » Subject 3C8 -> H/W -> Components
- Always re-read before sending

Demonstration Day is Friday April 14th

- Please be prepared to submit your deliverables in a timely manner.
- Demonstration timetable to be circulated later In the semester.

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Questions?

Lab Rules

- No eating or drinking in lab.
- No food or drink containers allowed in lab.
- No sitting on benches or window sills.
- Bags must be placed under the work benches.
- No radios, music players, newspapers in lab, turn off mobile phones or put on silent.

Lab Rules Coronavirus

- No eating or drinking in lab
- Register using the lab application:
  - https://www.tcd.ie/eleceng/covid19/
- Wipe down area before starting
- Wipe down area after session
- Wear a mask
- Do not touch items unnecessarily
- Leave when session is over, do not loiter

Lab Rules

- **Responsibility is on you to ensure your attendance is recorded**

- **After completion of work you must:**
  - Log off
  - Switch off any equipment you have been using.
  - Return all equipment, cables, leads and accessories to the proper storage locations, in an orderly manner.
  - Leave work area clean and tidy, all wires, solder etc. must be removed from work benches.
  - Chairs must be left under the bench in a tidy position.
  - Dispose of all rubbish.

Lab Rules

•Software on PCs

- Do not, under any circumstances, load software onto the hard disk such as games, plugins, applications, graphics.
- Do not modify software on the hard disk or modify background, wallpaper pictures, screen savers.
- Do not view or down load material from the internet which may cause offence to others.
- Storage space on local networks must only be used to store data and programs generated as part of an assignment.
- Keep your password secret. You are responsible for your own backups.

Lab Rules

•General Lab Safety

- All accidents must be reported to supervising lecturer or lab technician.
- Do not, under any circumstances, attempt to remove covers from equipment, fuses or plug tops.
- Report any damage of equipment, tools or leads to your lecturer or lab technician.
- Students must behave responsibly at all times when using the labs and equipment.
- Only use the type of tools provided by the college, no blades or knives.
- Wear goggles when soldering.
- If in doubt, ask!

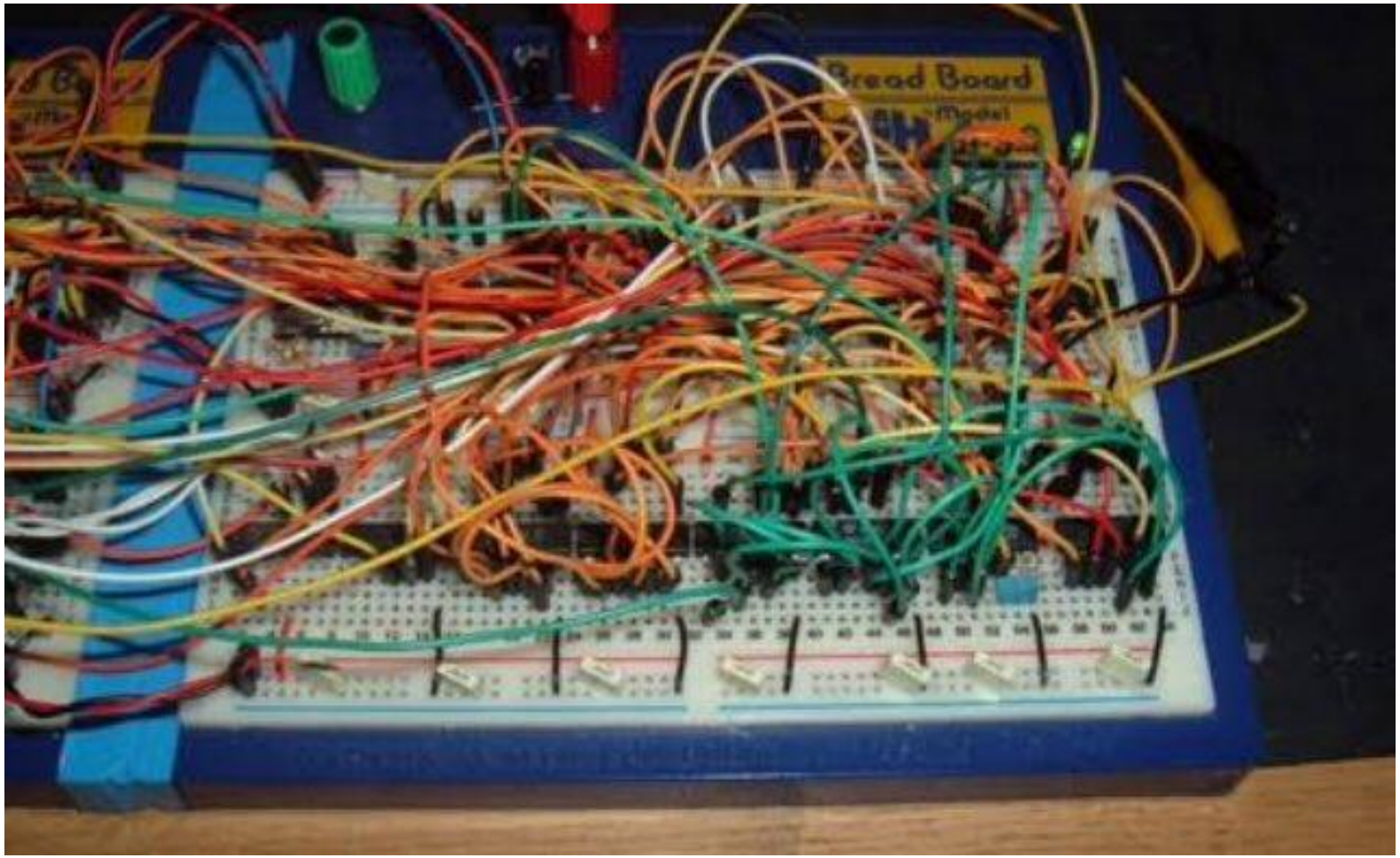
Lab Rules

- **If you have not been assigned a group, or listed in the incorrect stream, please contact eugene.orourke@tcd.ie using the following format:**
  - Subject -> Sub-Subject
  - Subject should be course code, e.g. 3E3 etc
  - Sub-Subject could be one of the following: Lab/Attendance/Non-Attendance/Hardware/Software/Report/Blackboard/Query/Absence

- **Response should include:**
  - Full Name:
  - Userid:
  - Student Number:
  - Group Number(if relevant):
  - Date (if relevant):
  - Issue to be resolved: - please ensure you have described the problem adequately and supply all pertinent information, error messages & screen snapshots
  - Architecture:PC vs MAC vs Linux
  - OS:
  - OS Version:

- **Before reporting forgotten password:**
  - Firstly, turn on caps key and try again.
  - Secondly, if you have any special characters(@, $, %, &) in your password, check to see if the keyboard is reflecting them properly using the username field, e.g. @ on English keyboard is " on US keyboard

Bad Breadboard



Fire Safety Notice

Where is the nearest fire exit?

Leave via top of lab, go left through firedoor, exit is on left. Follow exit signs. Final exit is onto Pearse Street. Make your way to the assembly point.

Where is the Assembly point?

Assembly point is the Rugby Pitch carpark. Please make your way there regardless of which exit you take. A roll-call or count may need to be taken.