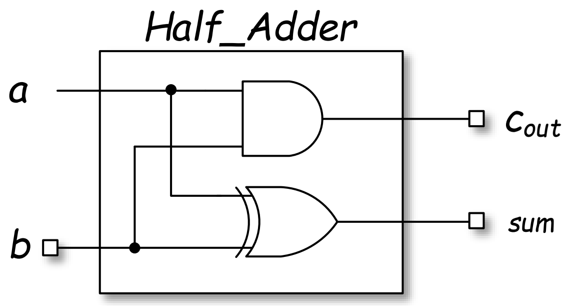
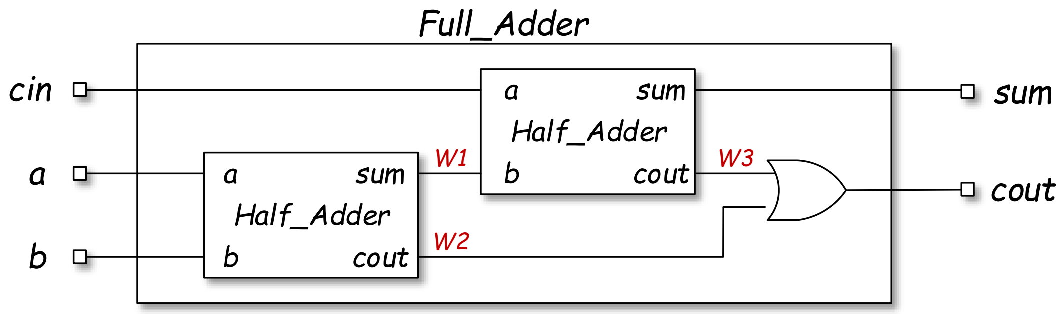
Digital System Design

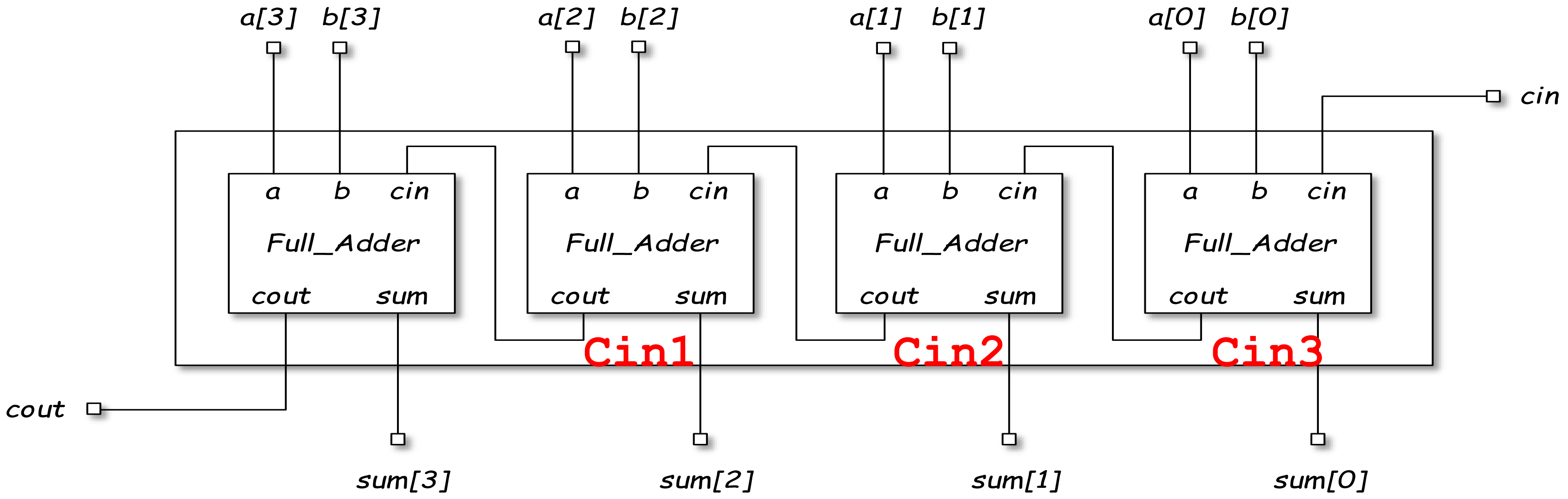
Hajar Falahati

hfalahati@ipm.ir
hfalahati@ce.sharif.edu

A 4-bit Full Adder

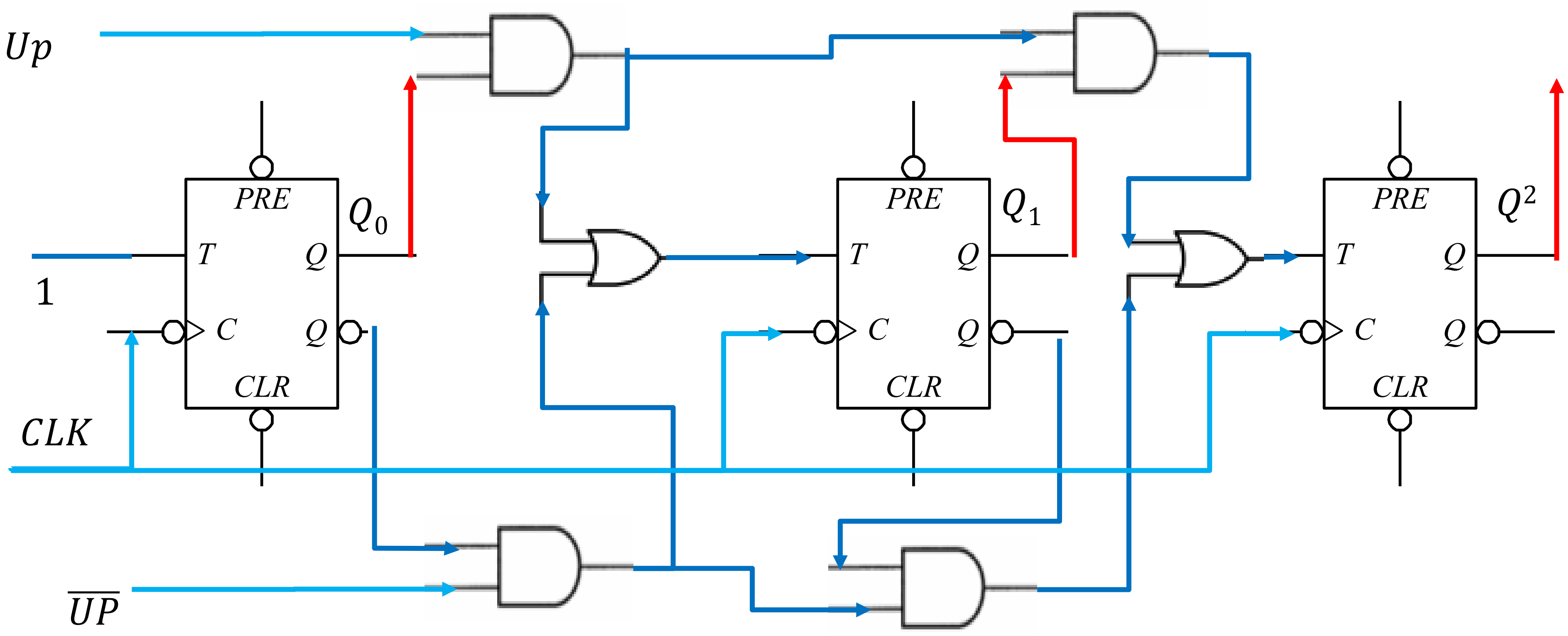






A 3-bit Counter

$T_0 = 1$ $T_1 = (Q_0 Up) + \overline{Q_0} . \overline{UP}$ $T_2 = (Q_0 Q_1 Up) + \overline{Q_0} . \overline{Q_1} . \overline{UP}$



Outline

- Why CAD?
    - What is wrong with traditional digital design?
    - Difficulties?
    - Solutions!

Digital System Design

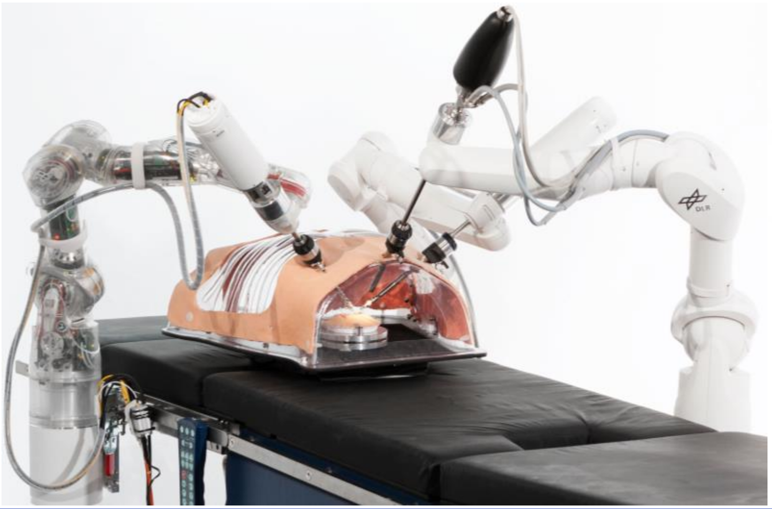
System Design

- You are **expert** in designing 🤓
  - Combinational circuits
  - Sequential circuits

- Design step
  - Model
  - Truth table
  - State diagram
  - Excitation table
  - K-map
  - Design

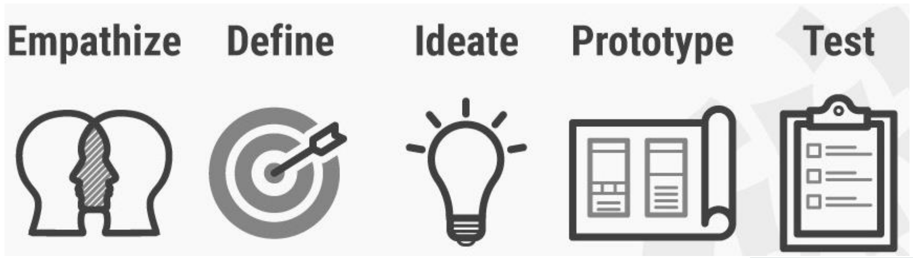
How Expert Are You?

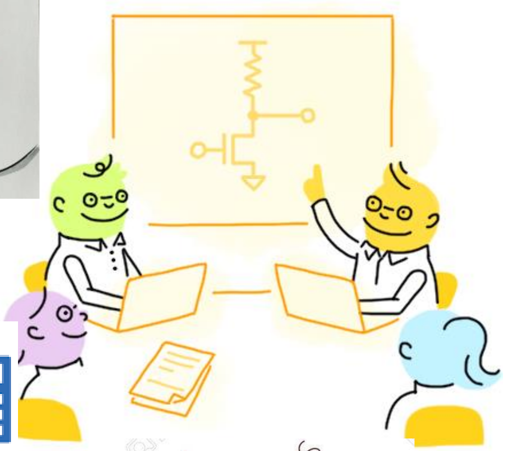
- Can you design every problem? 🤔
  - Elevator control
  - Autopilot system
  - Surgery robots



Any Challenges?

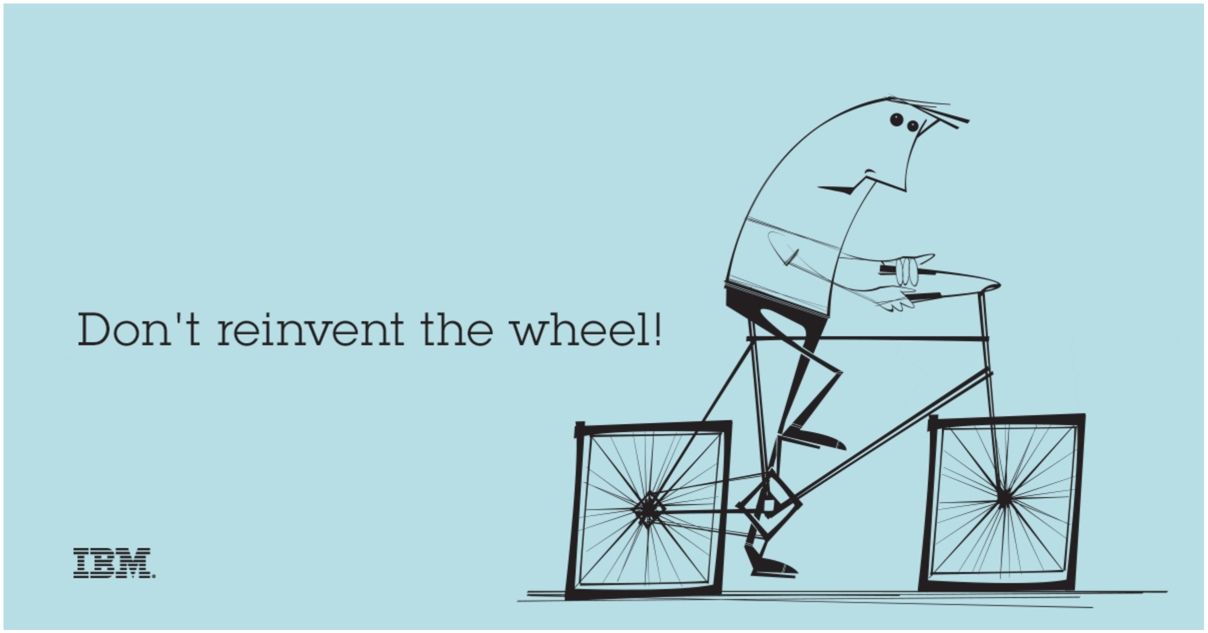
- Long prototyping time
- Long design time
- Human effort
- High risk
- Hard to detect errors
- Hard to verify
- Long time to market
- High cost
- Hard to update
- Hard to service
- Poor reusability





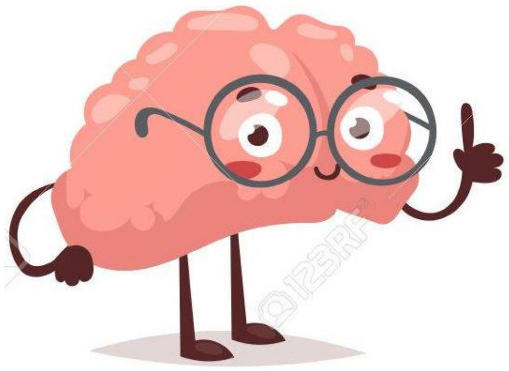
Any Solutions?

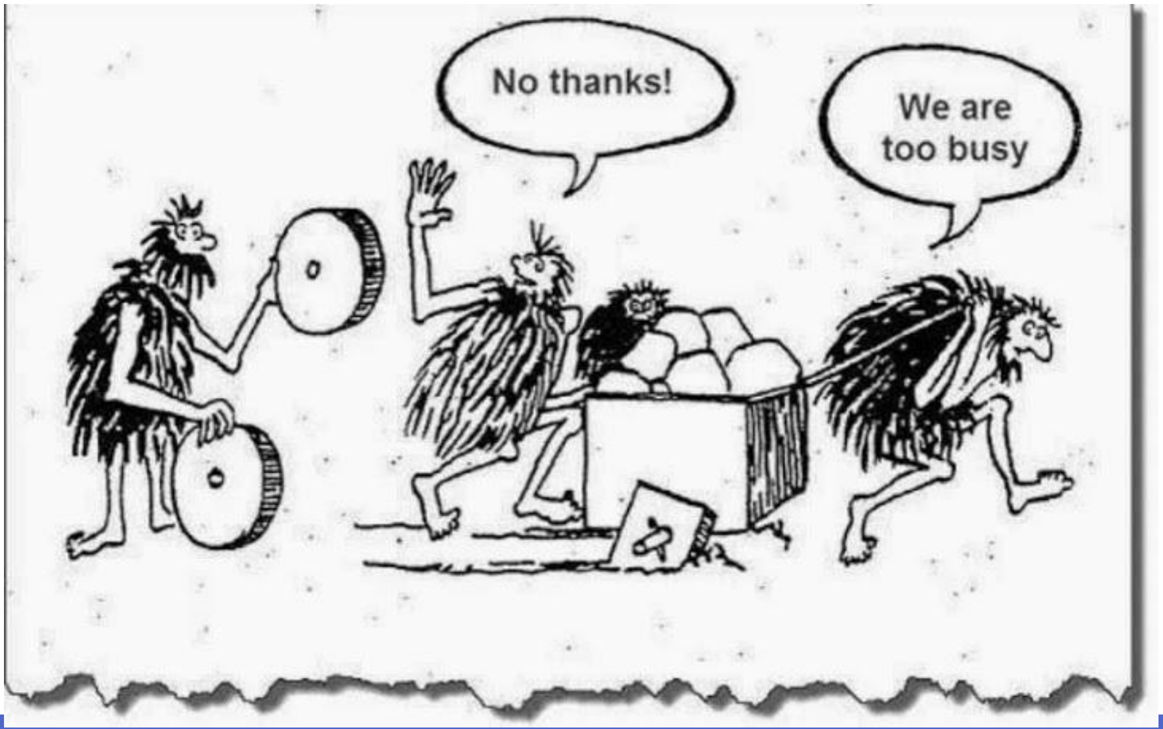
- Do we have the **same problem** in **other domains**, e.g., **software design**?



Any Solutions?

- Do we have the **same problem** in **other domains**, e.g., **software design**?
- How do **software programmers solve** the **problem**?



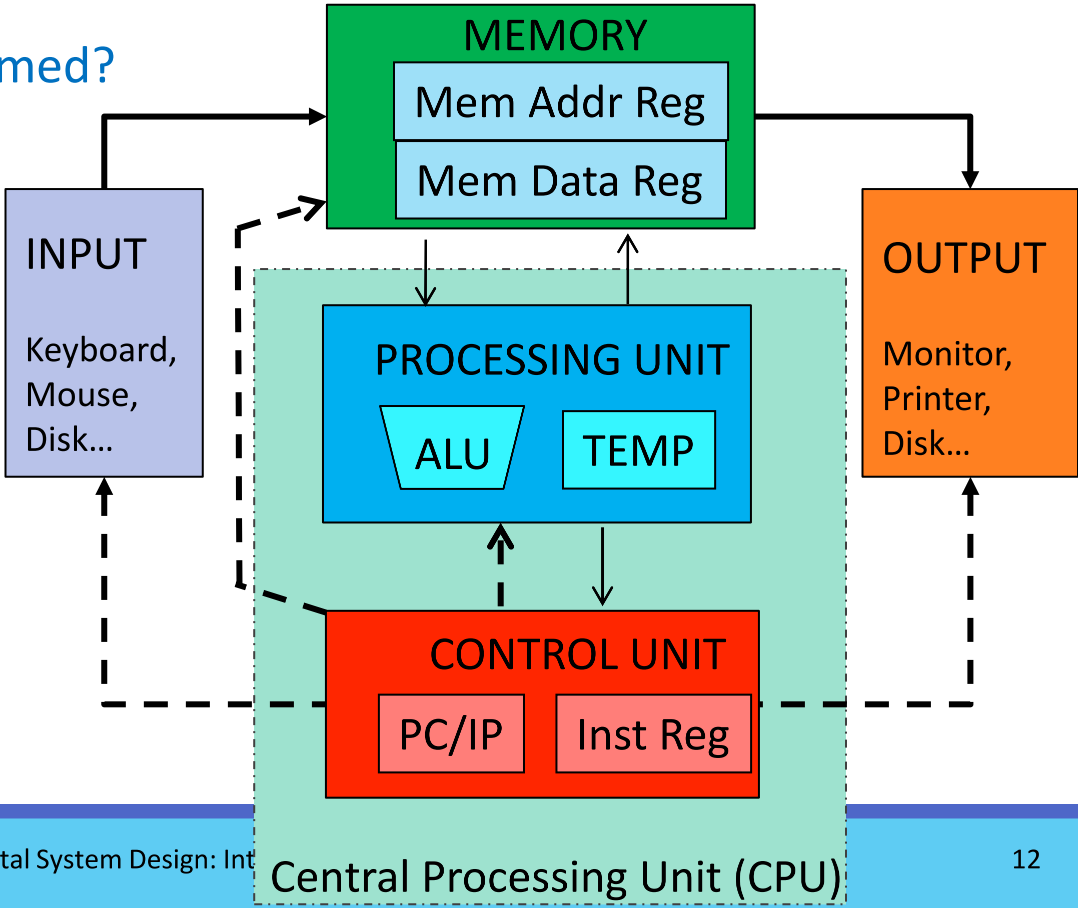


Programming in SW Domain

Programming

- Computers only understand **0's** and **1's**!

- How can we make it programmed?
  - Machine language!

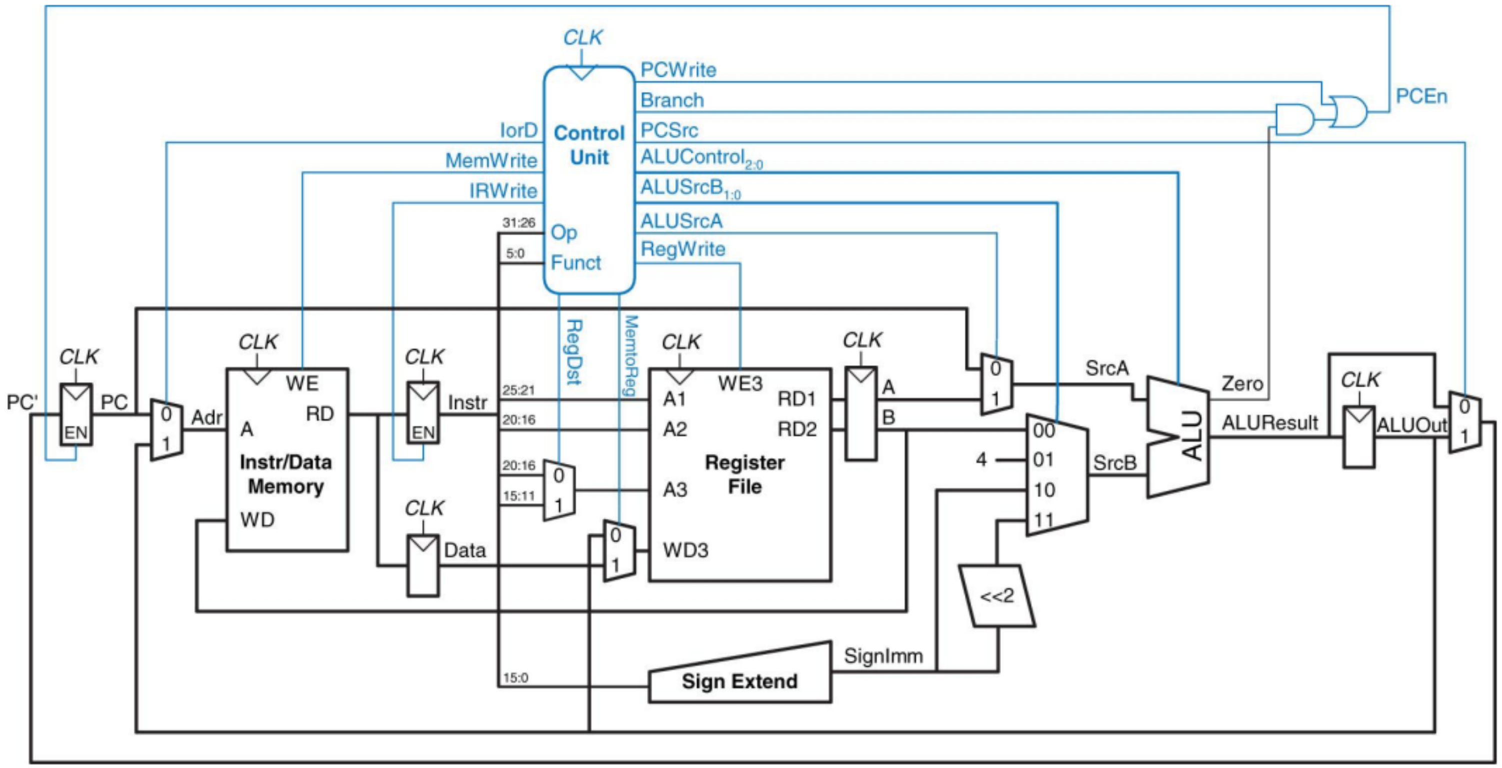


Programming

- Machine language
  - Binary representation of instructions

Computer
Machine code
01001000 01100101
01101100 01101100
01101111 00100001
Human
Human language
Hello!

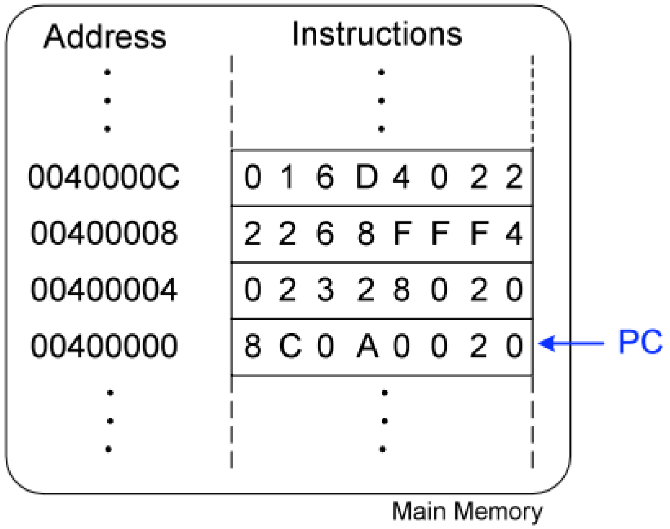
MIPS Processor



Machine Language

- So hard!

Stored Program



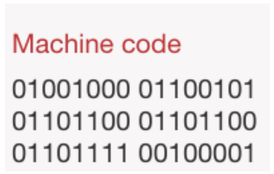
Machine Code

```
0x8C0A0020
0x02328020
0x2268FFF4
0x016D4022
```

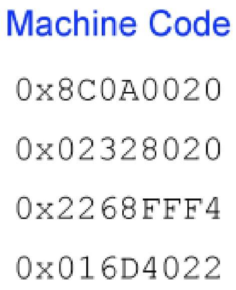
SW Vs. HW Domains:1

Software domain

- A bitstream of 0's and 1's


Machine code

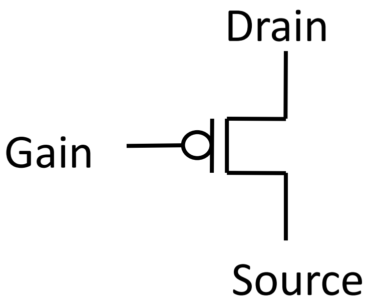
```
01001000 01100101
01101100 01101100
01101111 00100001
```


Machine Code

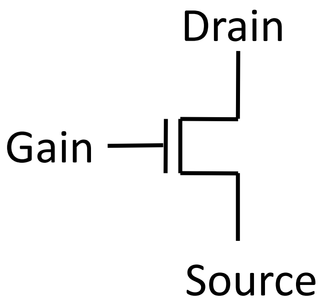
```
0x8C0A0020
0x02328020
0x2268FFF4
0x016D4022
```

Hardware domain

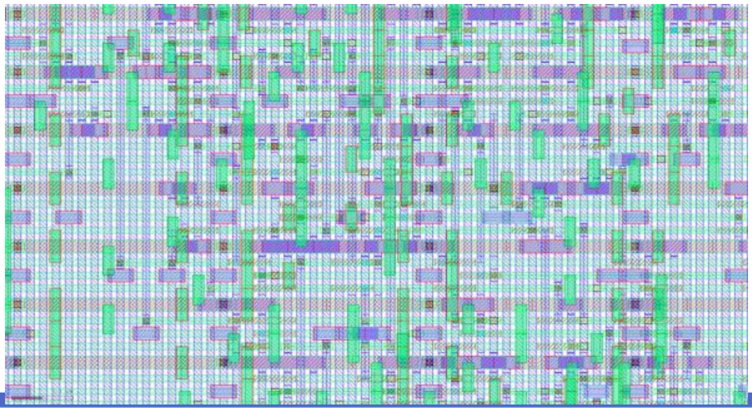
- Switch/ Transistor



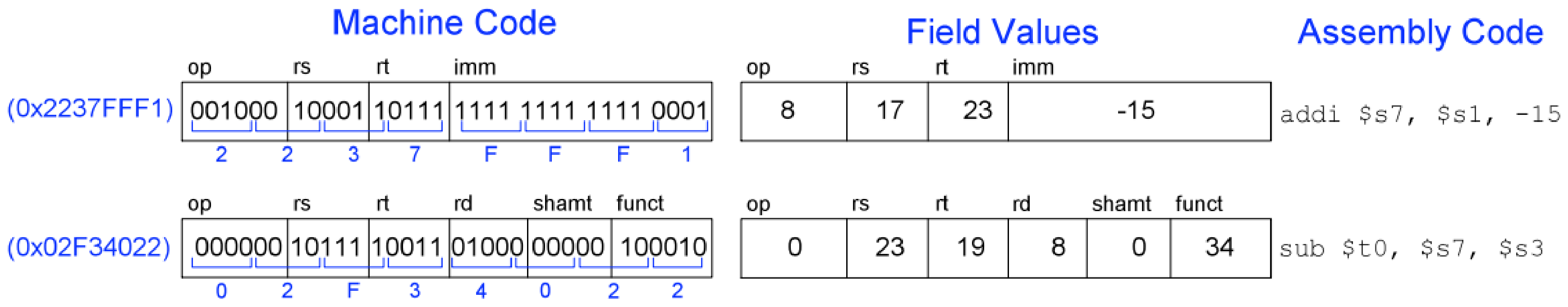
PMOS Transistor



NMOS Transistor



Assembly Language

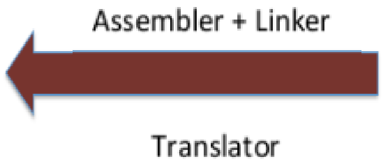


	Machine Code				Field Values						Assembly Code
	op	rs	rt	imm	op	rs	rt	imm			
(0x2237FFF1)	001000	10001	10111	1111 1111 1111 0001	8	17	23	-15			`addi $s7, $s1, -15`
	2 2		3 7	F F F 1							
	op	rs	rt	rd, shamt, funct	op	rs	rt	rd	shamt	funct	
(0x02F34022)	000000	10111	10011	01000 00000 100010	0	23	19	8	0	34	`sub $t0, $s7, $s3`
	0 2		F 3	4 0 2 2							

Machine Code

```
0x8C0A0020
0x02328020
0x2268FFF4
0x016D4022
```

addi $\$t_2$, \$0, 32
add $\$t_0$, $\$S_1$, S_2
subi $\$t_0$, $\$S_3$, -12
sub $\$t_0$, $\$t_3$, $\$t_5$



SW Vs. HW Domains:2

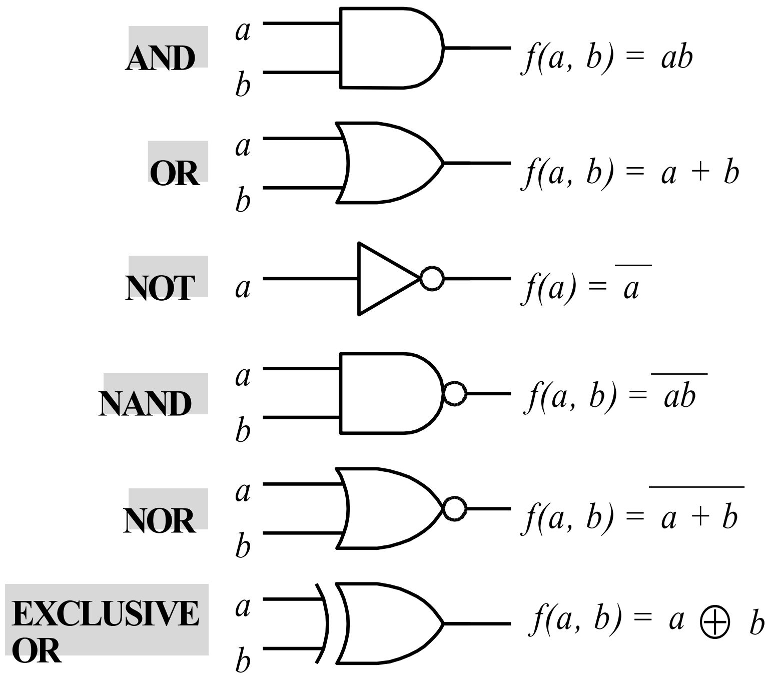
Software domain

- Assembly

```
addi  $t_2, $0, 32
add   $t_0, $S_1, S_2
subi  $t_0, $S_3, -12
sub  $t_0, $t_3, $t_5
```

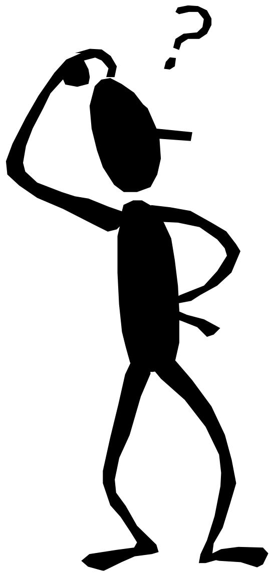
Hardware domain

- Logic gates



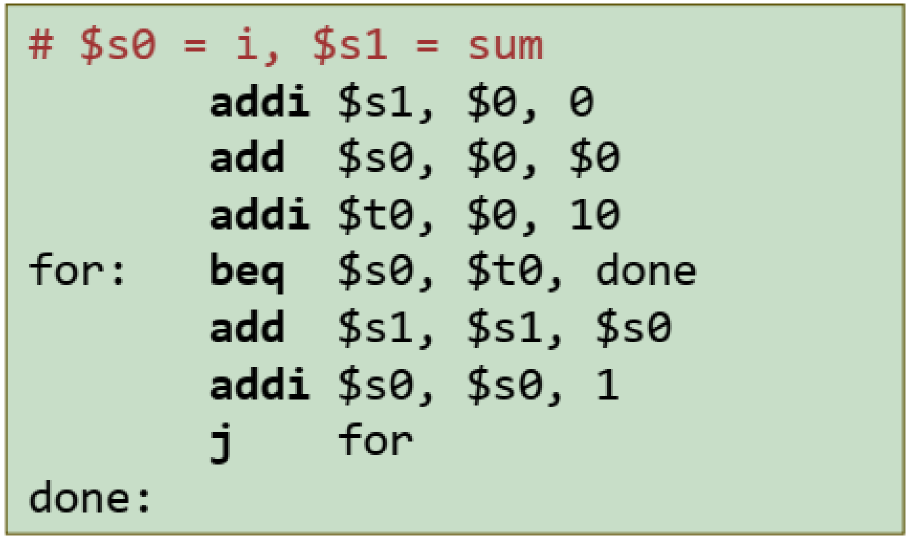
Sample Code

- Add the numbers from 0 to 9.



Sample Code: Assembly

- Add the numbers from 0 to 9.



```
# $s0 = i, $s1 = sum
        addi $s1, $0, 0
        add  $s0, $0, $0
        addi $t0, $0, 10
for:    beq  $s0, $t0, done
        add  $s1, $s1, $s0
        addi $s0, $s0, 1
        j    for
done:
```

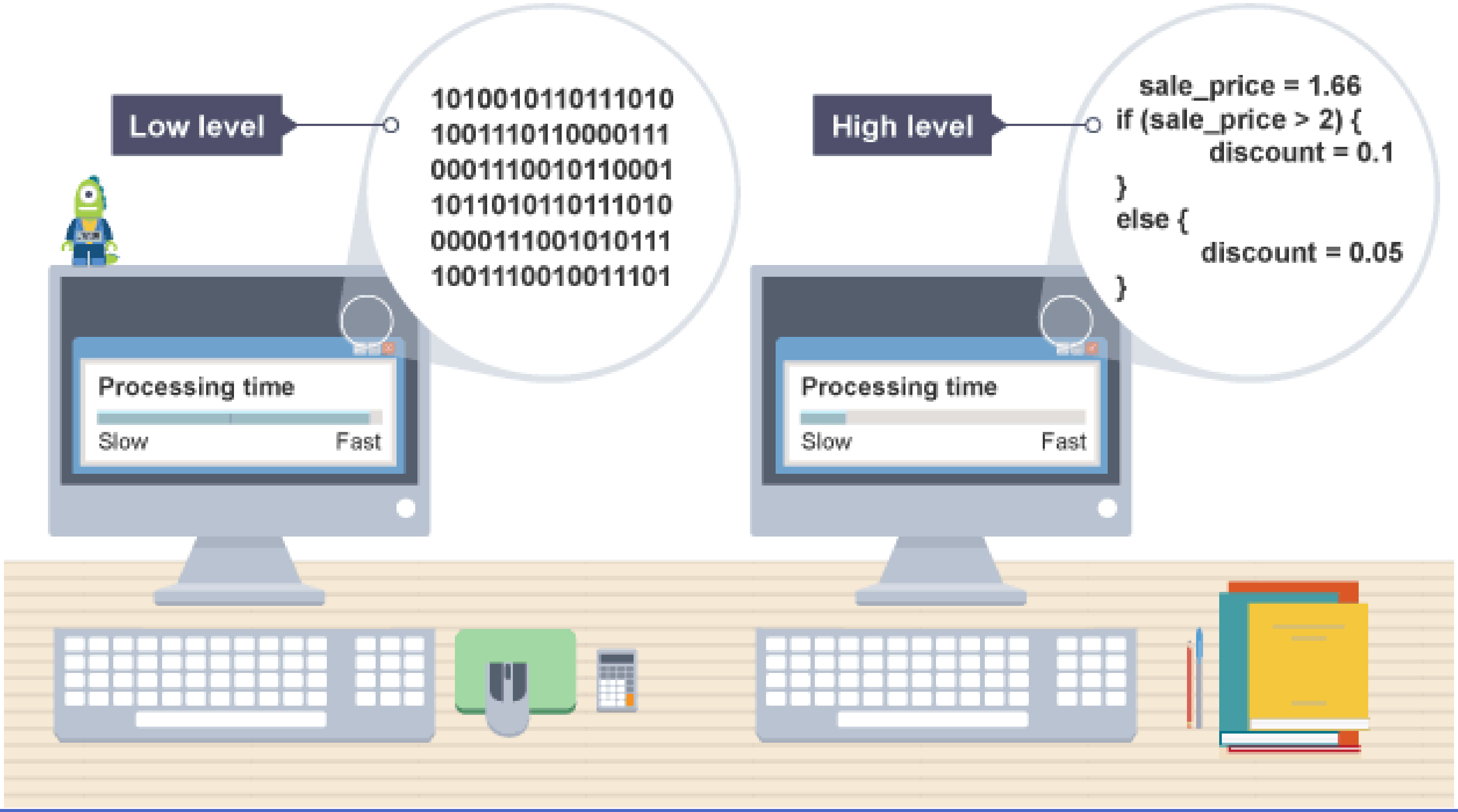
Sample Code: Assembly Vs. C

- Add the numbers from 0 to 9.

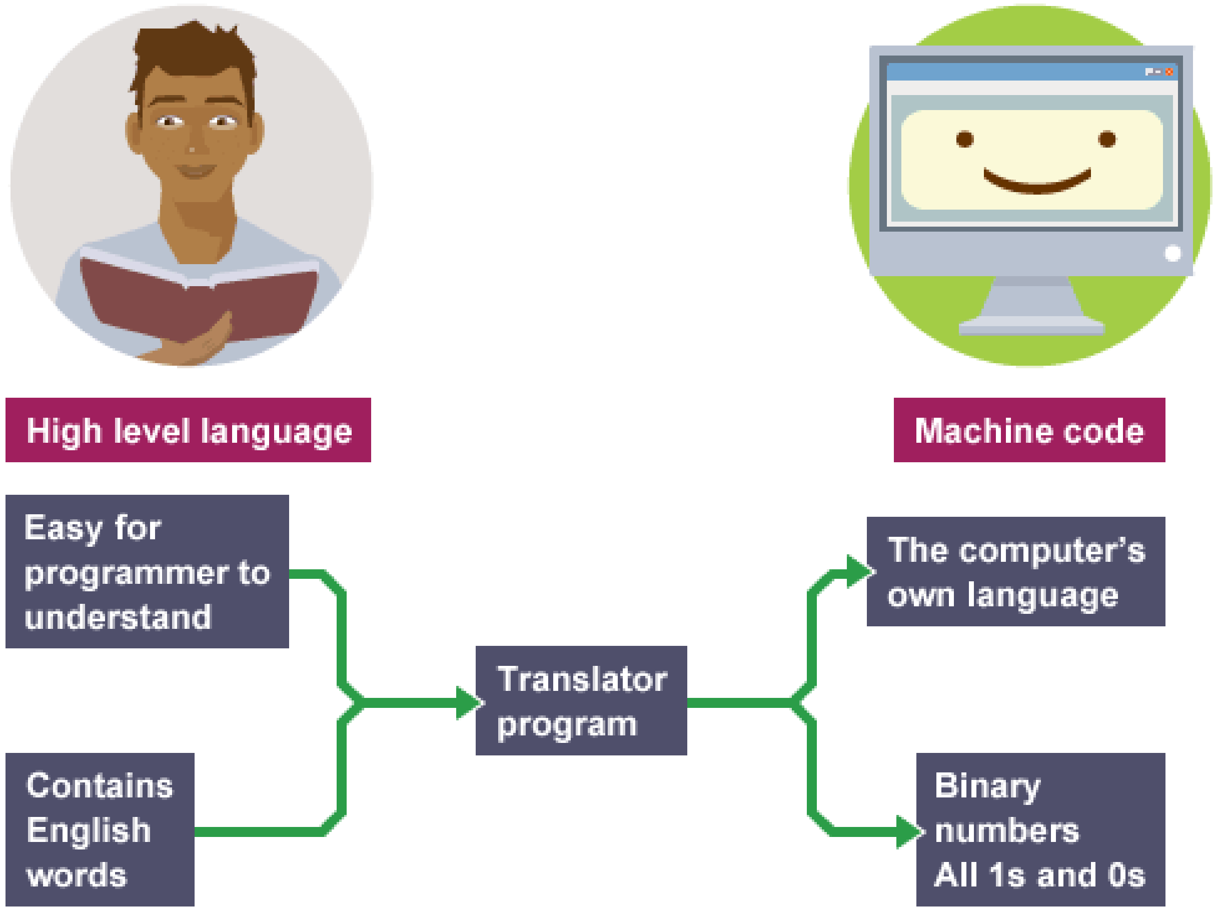
```
# $s0 = i, $s1 = sum
       addi $s1, $0, 0
       add  $s0, $0, $0
       addi $t0, $0, 10
for:   beq  $s0, $t0, done
       add  $s1, $s1, $s0
       addi $s0, $s0, 1
       j    for
done:
```

```
// add the numbers from 0 to 9
int sum = 0;
int i;

for (i = 0; i != 10; i = i+1) {
  sum = sum + i;
}
```

High Level Languages



High Level Languages


High level language
Machine code
Easy for programmer to understand
Contains English words
Translator program
The computer's own language
Binary numbers All 1s and 0s

SW Vs. HW Domains:3

Software domain

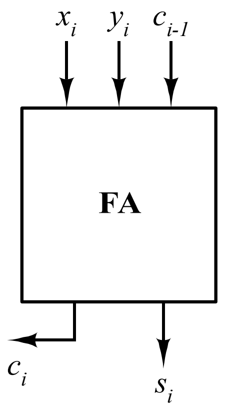
- High Level Language

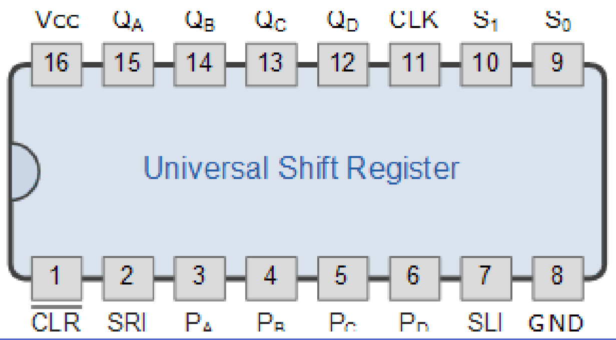
C = A + B

C >> 2

Hardware domain

- Logic blocks





HW Domains: Solutions

- Can we apply the **same solution** as software programming?
- Lets consider an example!

SW Vs. HW Domains:4

Software domain

- High level language

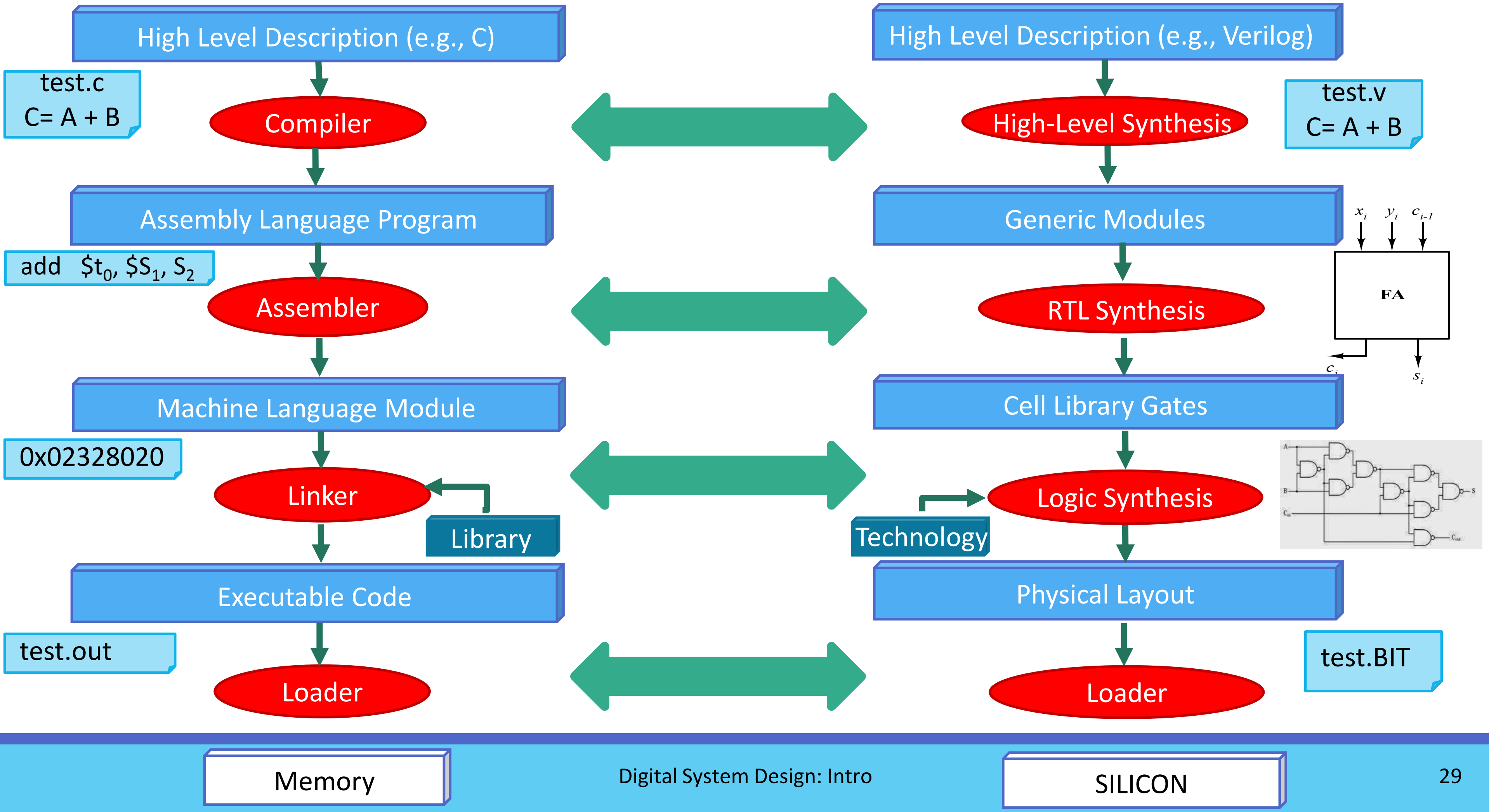
```
int main () {
   int A [8], B[8];
   int C[8];

   int i;
   for (int i=0; i<8; i++)
  {
        C[i] = A[i]+ B[i];
        C[i] >>2;
  }
}
```

Hardware domain

- Hardware description language (HDL)

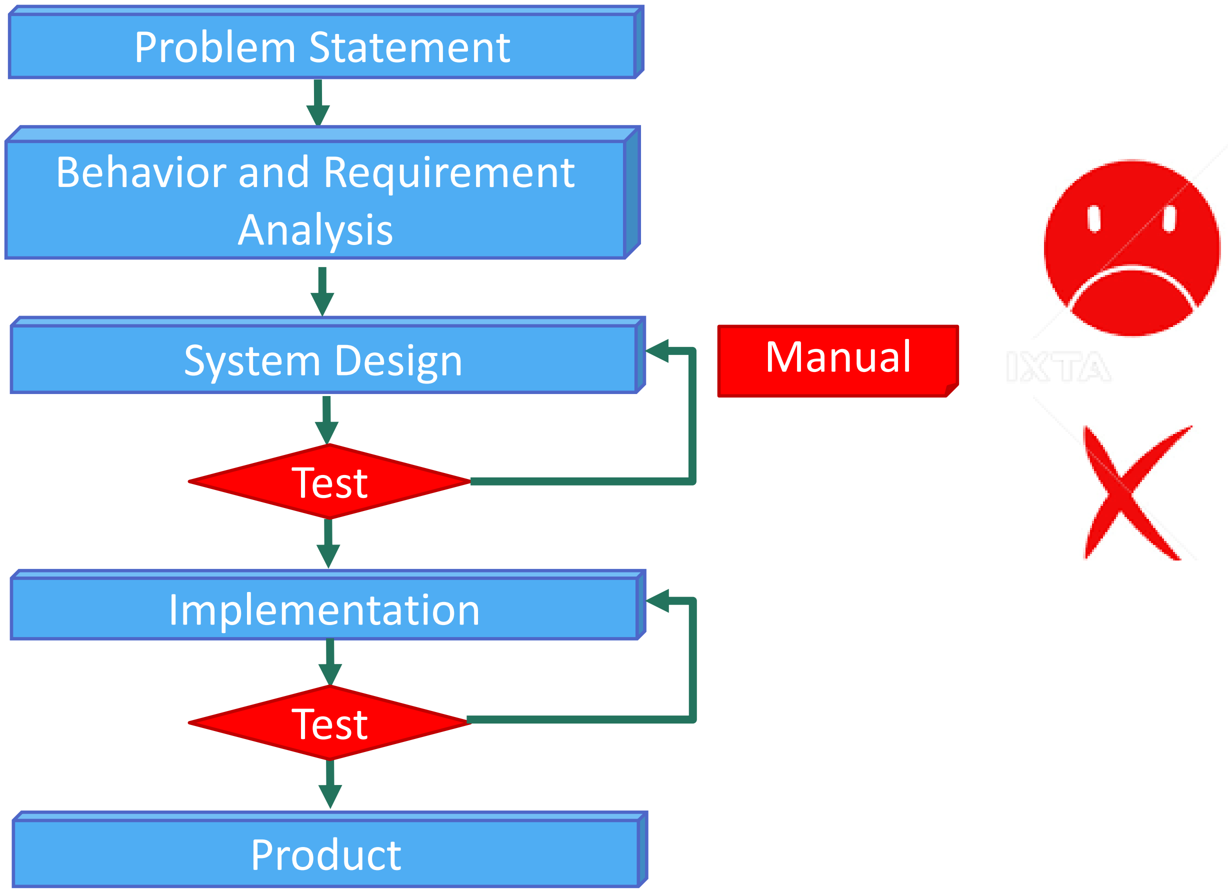
```
module main(A, B, C);
   input    [7:0] A, [7:0]B;
   output [7:0] C;

    integer i;
   for ( i=0; i<8; i = i+1)
     begin
        C[i] = A[i]+ B[i];
        C[i] >>2;
     end
endmodule
```

SW Vs. HW Domains:5

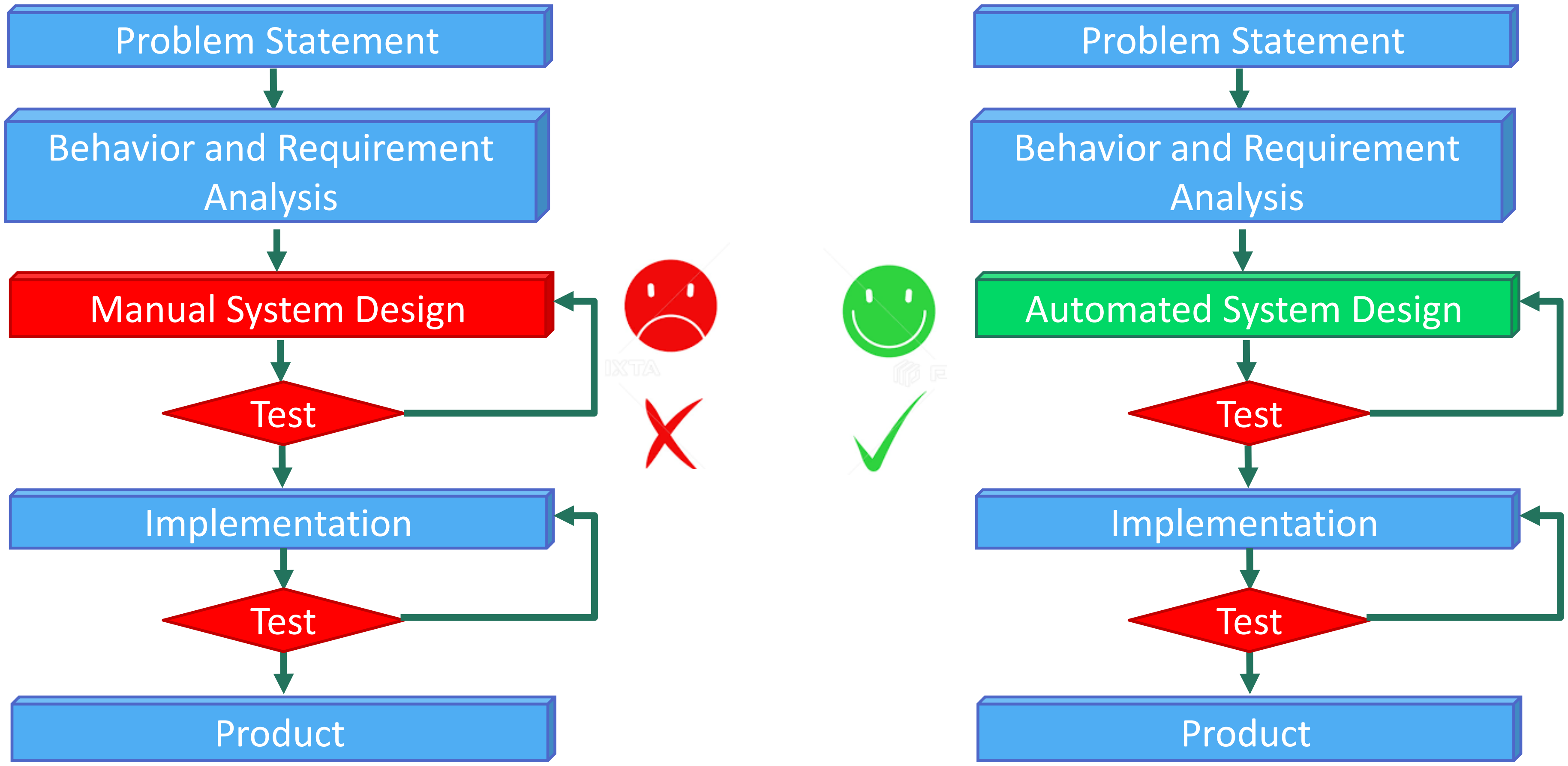


Problem Solved!

Design Flow


Problem Statement
Behavior and Requirement Analysis
System Design
Test
Implementation
Test
Product
Manual

Design Flow:
Manual Vs. Automated



Thank You