- **Explain briefly how the various characteristics of an embedded system can be fulfilled. Which methods and techniques are used?**

VARIOUS CHARACTERISTICS OF A MICROCONTROLLER:

1. **Reactive systems:**

Because of being non-terminating, it can perform faster reaction continuously interactingwith the environment, give a stimulus/response interaction

2. **Real-time systems:**

Performance of system behavior depends not only on the logical computational result,but also timely response of functional behaviors.

3. **Continuous/discrete/hybrid systems:**
-Continuous: required input/output behavior is specified using differential equations, numericalsolver.
-Discrete: State transition systems.
-Hybrid: Continuous and discrete elements „ Either independent or interrelated.

4. **Embedded systems:**

It is a software embedded in a technical system that interacts with the physicalcomponents (e.g., sensors) to control specific hardware.

5. **Dependable systems:**

Reliability on the system in two aspects:

- It performs according to its service specification.
- The system avoids hazards.

6. **Distributed systems:**

Multiple nodes/autonomous computers linked by a network. Communication andcoordination via message passing.

- **Give an overview of the attributes of dependability andshow how they influence each other.**

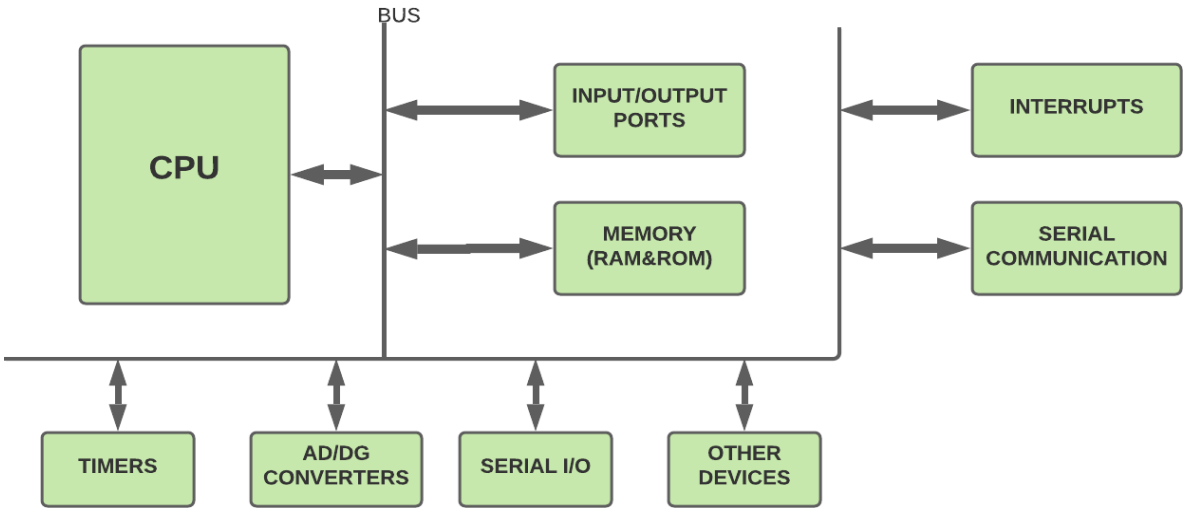
Reliability - dependability with respect to continuity of service. The ability of the system tofunction correctly (services as specified.) over a given period of time.

Availability - dependability with respect to readiness of the system to work perfectly at any given time or when requested.

Safety- Avoidance of Danger which the system may cause towards Human life or to environment.

Security - Prevention of unauthorized access and/or handling of information's. The ability of the system to protect itself against accidental or deliberate intrusion.

❖ **Give a schematic overview on the main elements of a microcontroller.**



❖ **Which processors are typically used for microcontrollers? Given an overview of the characteristics (like: #address bits, CPU frequency, # memory, …)**

A **microcontroller** is a small computer on a **single** integrated circuit chip. A **microcontroller** typically **contains one** or **more processor** cores, along with additional peripherals (memory, serial interface, timer, programmable I/O peripherals, etc.).

Processors typically used for micro-controllers:

- **General purpose microprocessor:** X86, PowerPC, … +processor board
- **Highly integrated microprocessor:** additional I/O on the chip

- **Single-chip microprocessor:** I/O, Rom, RAM
- **Digital signal processor:** extremely high throughput, optimized for numerical operations
- **Mixed-signal processor:** direct interface to analogous and digital signals (usually low-cost)

❖ **Implement a 2-bit (branch) predictor given by the state diagram in the lecture**

1) In C-Code

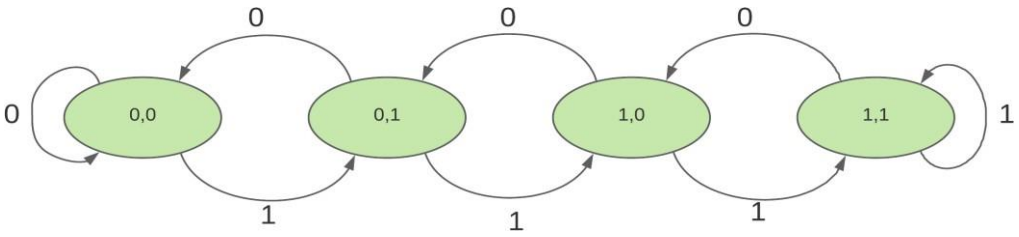
2) As a circuit (use the known memory elements from computer science 1)

Two Bit Predicator.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings

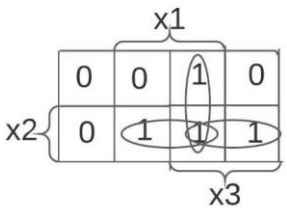
Start here | Two Bit Predicator.c

```
#define ST00 100
#define ST01 200
#define ST10 300
#define ST11 400

int status=ST00;

int main()
{
int NewInput;
for(;;)
{
printf("Enter New Input:1(Taken)/0(Not Taken)");
scanf("%d",&NewInput);
switch(status)
{
case ST00:
if(NewInput==1)
status=ST01;
break;
case ST01:
if(NewInput==1)
status=ST10;
else
status=ST00;
break;
case ST10:
if(NewInput==1)
status=ST11;
else
status=ST01;
break;
case ST11:
if(NewInput==0)
status=ST10;
break;
default:
break;
}
printf("\nNew Status=%d\n",status);
}
}
```



x1	x2	x3	x1n	X2n
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

x1n

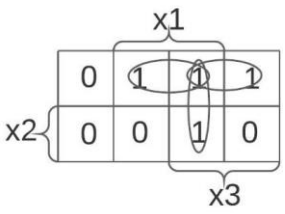


x1∧ x2

x1∧x3

x2∧x3

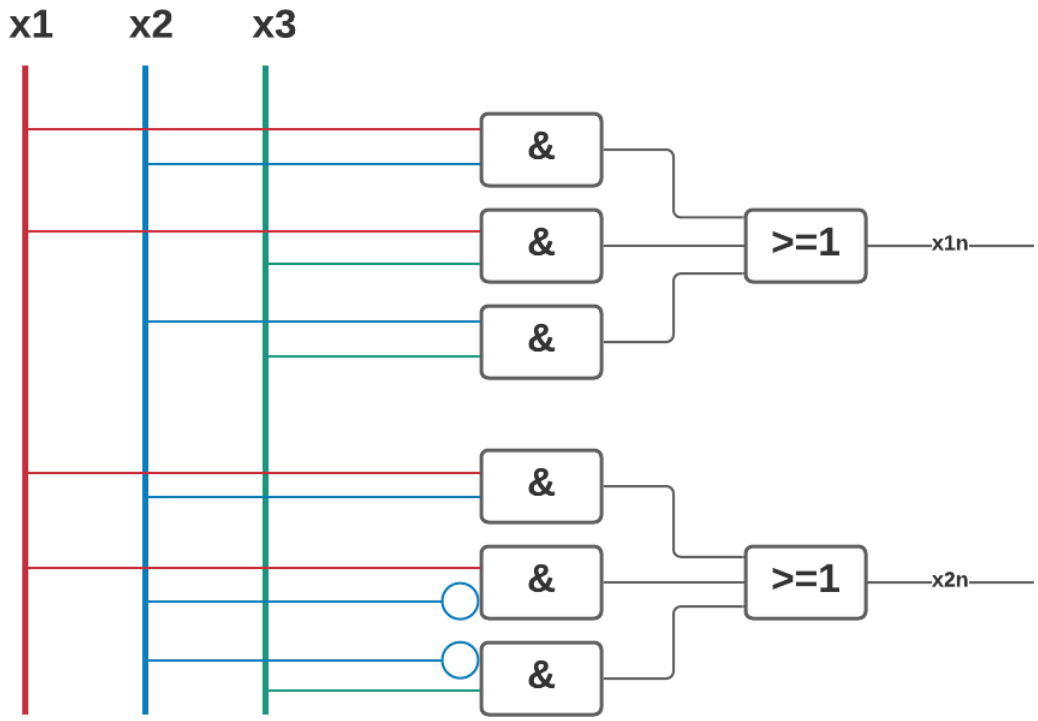
x2n

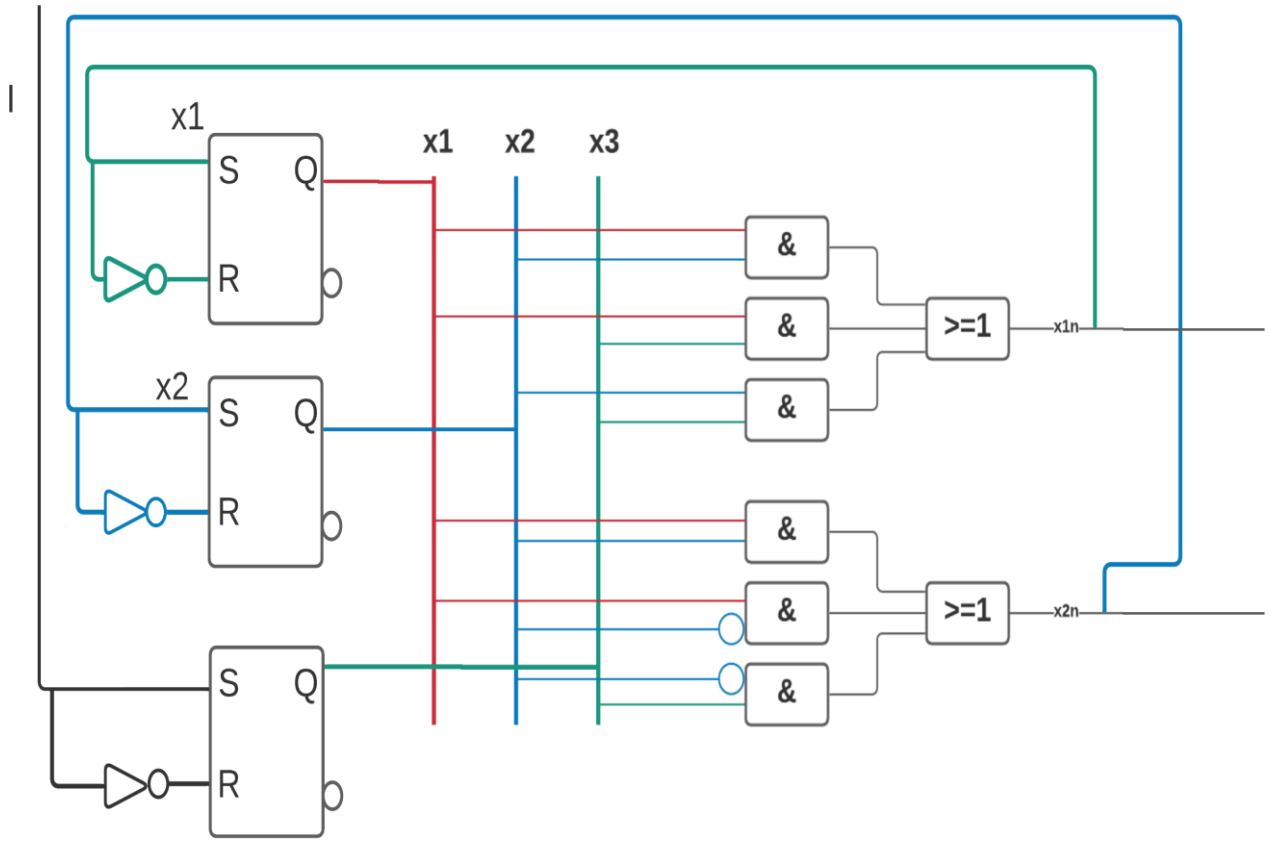


x1∧x3

x1∧ $\overline{x2}$

$\overline{x2}$ ∧x3


x1
x2
x3
&
&
&
>=1
x1n
&
&
&
>=1
x2n


I
x1
S
Q
R
x2
S
Q
R
S
Q
R
x1
x2
x3
&
&
&
>=1
x1n
&
&
&
>=1
x2n

- **How can we implement reactive system as a C program and Arduino program?**

As C program:

reactive.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Start here X reactive.c X

```
1  int main()
2  {
3  int answer;
4  char ch;
5  int num1,num2,a,m,s;
6  float d;
7  do{
8  printf("\nEnter the first number:");
9  scanf("%d",&num1);
10 printf("\nEnter the second number:");
11 scanf("%d",&num2);
12 a=num1+num2;
13 m=num1*num2;
14 s=num1-num2;
15 d-(float)(num1/num2);
16 printf("\nEnter your choice\nFor Additional Type A\nFor Multiplication Type M\nFor Division Type D\nFor Subtraction Type S: \n");
17 scanf("%c",&ch);
18 switch(ch)
19 {
20 case 'A':printf("\nThe Addition of the number is= %d\n",a);
21 break;
22 case 'M':printf("\nThe Multiplication of the number is=%d\n",m);
23 break;
24 case 'S':printf("\nThe Subtraction of the number is=%d\n",s);
25 break;
26 case 'D':printf("\nThe division of the number is =%d\n",d);
27 break;
28 default: printf("\nInvalid Entry");
29 break;
30 }
31 printf("Please press 0 to continue and any digit to exit\n",answer);
32 scanf("%d",&answer);
33 }
34 while (answer==0);
35 return 0;
36
37 }
38
```

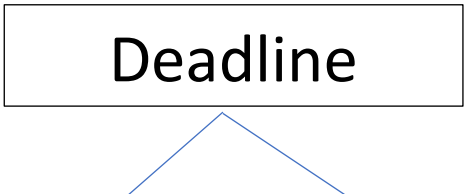
As Arduino Program:

Reactive_System.ino | ReadMe.adoc

```
1  int i=0;
2
3  void setup() {
4    Serial.begin(9600);
5  }
6
7  void loop() {
8    Serial.println(i);
9    i++;
10 }
11
```

- **What is the meaning of deadline? How can we ensure to be within deadline? (Examples, tools etc.)**

Deadlines in a real time system represent the time at which specific tasks have to be completed. Each task in a real time system will have its own deadline. It is the time by which a system must produce an output.



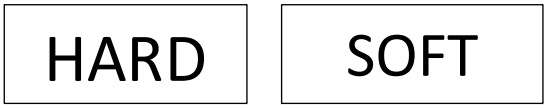


Figure1.1: Types of Deadline

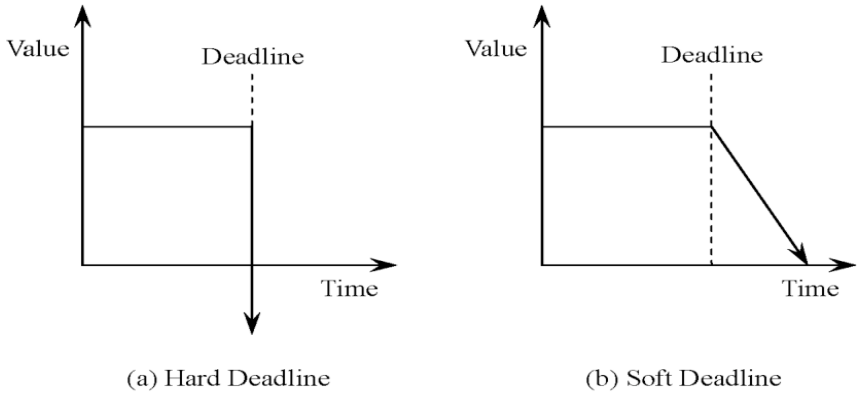


Figure1.2: Deadline in Hard real-time system and soft real-time system

Hard: Penalty due to missing deadline is a higher order of magnitude than the reward in meeting the deadline missing.

Soft: Penalty due to missing deadline is a lesser magnitude than reward in meeting the deadline missing.

Examples:

1.1 Engine control system, pacemakers, Avionic systems

A car engine control system is a hard real time system because a delayed signal may cause engine failure or an accident.

Such strong guarantees are required of systems for which not reacting in a certain interval of time would cause great loss in some manner especially damaging the surroundings physically or threatening human lives.

1.2 Airlines database, Mobile phone, Live audio-video systems.

The flight plans must be kept reasonably current but can operate to a latency of

seconds.

Soft deadline system violation of constraints results in degraded quality, but the system can continue to operate.

Reactive procedures are built from reactive statements and indicate the way to execute them. Reactive procedure instants are identified with their calls.

To implement the reactive-procedure notion, consider the following C function that prints ‘hello, world’ at each call:

```
hello(){
printf( " hello, world\n " );
}
```

suppose one wants to print ‘hello, world’ during the first call and ‘I repeat: hello, world’ during the second call. One writes in RC

```
rproc Hello(){
printf( " hello, world\n " );
}
```

- **Details about embedded systems including different hardware platforms.**

Embedded system can be defined as “A combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a dedicated function. In some cases, embedded systems are part of a larger system or product, as in the case of an antilock braking system in a car.”
Embedded systems are special purpose systems where the system is fully encapsulated by the device it controls. Such system performs one or more predefined tasks, usually with very specific requirements.
Essential Parts of an Embedded System

- Memory
- CPU
- Hardwired unit (Application of specific logic timers, A/D,D/A)
- Sensors
- Actuators

Possible Hardware Platforms

- General purpose microprocessor: CISC are Intel 386, RISC are IBM RS6000, EPIC is IA-64 (Intel Architecture-64), etc.

- Highly integrated microprocessor: Microcontrollers, microprocessors, and FPGAs

- Single-chip microcomputer: I/O, Rom, RAM, …
- Single-chip microcontroller: microcomputer with Realtime clock, A/D and D/A converters, …
- Digital signal processor: Thermometer, Analog/digital clock, smartwatch.
- Mixed-signal processor: The system may be implemented in the form of a printed circuit board, hybrid microcircuit, or a single integrated circuit chip.

.