Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Institution of Manipal University

SEVENTH SEMESTER B.Tech. (E & C) DEGREE END SEMESTER EXAMINATION
NOV/DEC 2017
SUBJECT: ADVANCED EMBEDDED SYSTEM DESIGN (ECE - 4001)

TIME: 3 HOURS **MAX. MARKS: 50**

Instructions to candidates
- Answer **ALL** questions.
- Missing data may be suitably assumed.

1A. Assume the features of an embedded system are given in the form of technical tasks *T0, T1, T2, T3, T4* and *T5*. The flow of execution of these tasks is shown in **Fig.1.1**. The execution time(T) and power consumption(P) of each task by various processing elements is given in **Table 1.1**.

i. Explain the concept of co-synthesis in a typical embedded system design flow.
ii. Find suitable application mapping to design the system with power consumption less than 30 mWatts by drawing activity scheduling graph.
iii. Find suitable application mapping to design the system with total processing time less than 20msec by drawing activity scheduling graph.

1B. Write the differences among the following

i. MicroBlaze processor and ARM cortex-A9 processor
ii. Zynq SoC and Programmable SoC
iii. Executable linkable format file and Bit file

1C. A young engineer is in confusion of selecting proper processing element to implement the following operation:

$$E = A \; X \; B,$$
$$F = C \div D,$$
$$G = E + F$$

[where A,B,C,D,E,F are registers of appropriate size]

If high speed & low cost are his primary objectives then which one you suggest among GPP, FPGA and ZYNQ SoC to get optimum solution. Give reason.

(5+3+2)

2A. Below are the PSoC Creator macros for Watch Dog Timer,

CySysWdtWriteCascade(0x08);
CySysWdtWriteMatch(argument1, argument2);
CySysWdtWriteMode(argument3, argument4);
CySysWdtWriteClearOnMatch(argument5, argument6);
CySysWdtEnable(argument7);

Utilize the given macros, configure them appropriately and write pseudo code to do the following:

i. To reset CPU for every 2 seconds

ii. To interrupt CPU for every 3.5 seconds

iii. Configure to generate an interrupt for every 4 seconds and reset if it is not serviced consecutively consecutively.

2B. In context of PSoC 4 interrupts, explain the use of following
   i. **<*instance_name*>_Start()** and **<*instance_name*>_StartEx()** API routines
   ii. **CyEnterCriticalSection()** API routine
   iii. Sticky bits

2C. Write the special features those Digital signal processor should have over other data processing elements.

(5+3+2)

3A. From the component catalogue of PSoC Creator, assume we have selected one *DigitalIn*, two *DigitalOut* components and named them as **SW**, **LED_GREEN**, **LED_RED** respectively. The two *DigitalOut* pins are configured in *Strong drive mode* with initial drive state as *LOGIC ZERO* as shown in **Fig.3.1**. When working with CY8CKIT-044, if the code in *main.c* is as shown in **Fig.3.2** then write and explain in detail the status of **LED_GREEN** and **LED_RED** pins under the following drive conditions of **SW** pin when it is not connected to any digital value externally
   i. Drive mode: *Resistive pull up*, Initial drive: *'0'*.
   ii. Drive mode: *Resistive pull down*, Initial drive: *'1'*.
   iii. Drive mode: *Strong*, Initial drive: *'0'*.
   iv. Drive mode: *Resistive pull up*, Initial drive: *'1'*, SW: connected to '0'
   v. Drive mode: *Resistive pull down*, Initial drive: *'0'*, SW: connected to '1'

3B. Assume the schematic in ***.Cysch*** file of PSoC creator is as shown in **Fig 3.3**. ***Pin_Start_Bootloader*** is configured to **Resistive Pull Up** mode with **Falling edge** interrupt and connected to P0[7] of CY8CKIT-044. ***Pin_LED*** is configured in **strong drive mode**. Bootloadable is linked to a Bootloader, which is configured to communicate with Host using I2C. Rest all are default.
Fill ***main.c*** file of PSoC Creator with appropriate code to do the following:
CPU should toggle ***Pin_LED*** continuously (with a period of 1 sec) as long as the switch on CY8CKIT-044 is not pressed. When switch pressed, CPU Should start executing Bootloader and it should remain doing the same until HOST update flash with new application.

3C. Draw the watch dog block diagram for PSoC 4200M devices.

(5+3+2)

4A. With the help of transition diagram, explain the power saving modes of PSoC 4.

4B. Draw the block diagram of PSoC4 UDB and explain the advantage of data-path availability in it. List out the methods available to embed functionality into PLDs and Datapath of PSoC UDBs.

4C. What is a Bootloader? Name any of the two communication protocols generally used to update the flash of PSoC4 with Bootloader. What will be the format of Bootloadable file used to update flash.

(5+3+2)

5A. Explain the configuration of Instruction, Registers, Input and Output windows of DP element of UDB Editor along with its control register to implement the state diagram shown in **Fig.5.1.**

5B. Assume the schematic in *.cysch* file of PSoC Creator is as shown in **Fig.5.2**. Write C code to convert analog input to digital and send the LSB of result through digital output pin *Pin_LED*.

5C. What is Dynamic Reconfiguration of PSoC? Explain with an example.

(5+3+2)

	GPP		DSP		FPGA		ASIC	
Task	***T(ms)***	***P(mw)***	***T(ms)***	***P(mw)***	***T(ms)***	***P(mw)***	***T(ms)***	***P(mw)***
T0	24.6	2.1	8.4	9.4	3.2	17.2	1.8	26.2
T1	7.2	9.7	9.7	7.2	17.6	2.8	14.8	7.0
T2	6.4	16.4	7.0	14.8	26.4	1.2	22.7	2.2
T3	26.2	1.8	18.4	2.4	9.4	8.4	8.8	8.8
T4	16.4	6.4	17.2	3.2	2.2	22.7	2.1	24.6
T5	6.4	16.4	1.2	26.4	2.8	17.6	2.4	18.4

T(ms)*-Time taken by each task to execute(in milli seconds).*
P(mw)*- Power consuming by each task (in milli Watts).*

Table 1.1

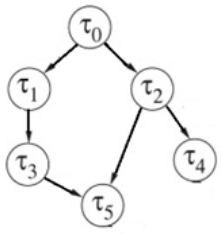

Fig.1.1

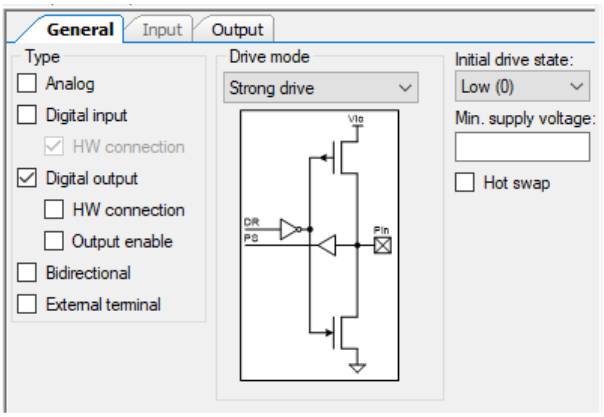

Fig.3.1

```
for(;;)
  {
    if (SW_Read()==1)
      {
      LED_GREEN_Write(~LED_GREEN_Read());
      CyDelay(500);
      }
    else if(SW_Read()==0)
      {
      LED_RED_Write(~LED_RED_Read());
      CyDelay(500);
      }
  }
```

Fig.3.2

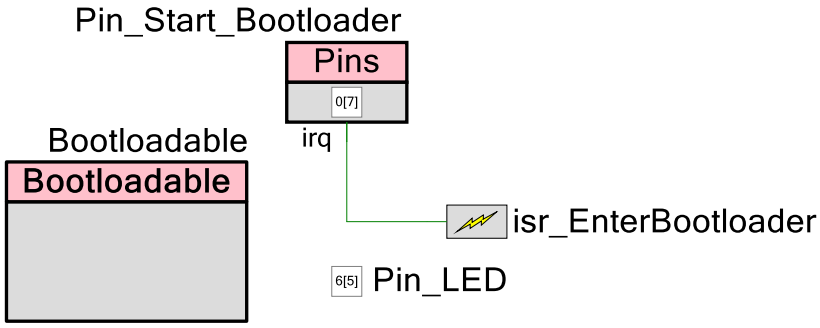

Fig.3.3

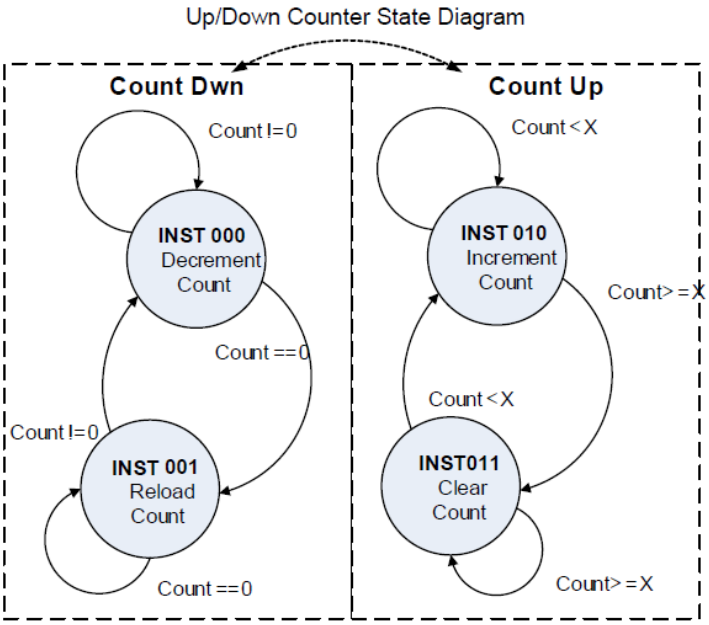
Up/Down Counter State Diagram
Count Dwn
Count Up
Count !=0
INST 000
Decrement
Count
Count ==0
Count !=0
INST 001
Reload
Count
Count ==0
Count < X
INST 010
Increment
Count
Count> = X
Count < X
INST011
Clear
Count
Count> = X

Fig. 5.1

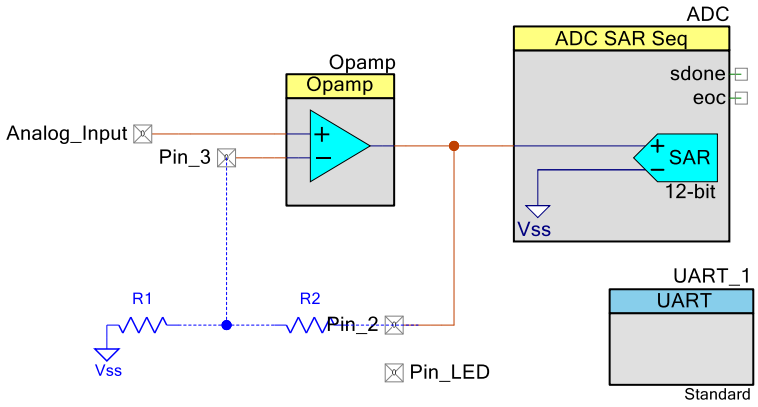
ADC
ADC SAR Seq
sdone
eoc
Opamp
Opamp
Analog_Input
Pin_3
SAR
12-bit
Vss
UART_1
UART
R1
R2
Pin_2
Vss
Pin_LED
Standard

Fig. 5.2