**Unive9rsidade Federal de Roraima Departamento de Ciência da Computação**

**Introdução a Sistemas Embarcados**

DISCIPLINA: INTRODUÇÃO A SISTEMAS EMBARCADOS

PROFESSOR: Herbert Oliveira Rocha

ALUNA: Larissa Santos Silva

Lista de Exercício 01

**[QUESTÃO 01]. Defina sistemas embarcados e apresente 3 exemplos com justificativa.**

O sistema embarcado ou sistema embutido é um microprocessador no qual o computador e completamente encapsulado ao sistema que ele controla. E são caracterizados pelas restrições de tempo e redução de energia.

Exemplo: Por exemplo, um Microondas contém um sistema incorporado que aceita Entrada do painel, controla o visor LCD e Liga e desliga os elementos de aquecimento que cozinham Comida.

Um sistema controlador de velocidade de cruzeiro em automóvel, regula a velocidade do carro percebendo as rotações do eixo dianteiro, o velocímetro, a chave de controle de cruzeiro, o acelerador e o pedal de freio é possui severas restrições temporais e outras relativas a desempenho e segurança

Máquina equipada com leitores de etiquetas inteligentes - Os quais devem, nos próximos anos, substituir os códigos de barra. Quando o usuário coloca suas roupas na máquina, os sensores lêem as etiquetas inteligentes das peças e avaliam se aquela lavagem pode ou não ser feita. Caso as cores ou tecidos sejam "incompatíveis", uma pequena tela indica que aquela operação de lavagem não será realizada e explica o motivo.

**[QUESTÃO 02]. Apresente uma comparação entre sistemas embarcados e sistemas tradicionais.**

Os sistemas tradicionais podem fazer um computador mais eficiente, pois não podem lidar com a questão central, ou seja, projeto embarcado trata com o sistema, não com computadores em si. Em sistemas embarcados trabalha-se mais com combinação de interfaces externas (sensores e atuadores) e algoritmos de controle são de maior importância. A CPU simplesmente existe para implementar essas funções.

**[QUESTÃO 03]. Descreva os que são restrições temporais e de consumo de energia**.

As restrições dos temporais são um sistema de tempo-real é um no qual a corretude depende não somente do resultado lógico, mas também do tempo em que tal resultado é produzido.

As restrições para o consumo de energia servir para consumir o mínimo de energia é assim prolongar a vida da bateria ou evitar a necessidade de um ventilador de arrefecimento

**[QUESTÃO 04]. O que é hardware/software codesign? Descreva as etapas (fases)?**

*Hardware/Software co-design* é o paradigma cujo principal objetivo é projetar sistemas digitais que satisfaçam às restrições de projeto utilizando componentes de prateleiras, e juntamente componentes de aplicação específica. Para um projeto mais completo e consistente é necessário que estes componentes heterogêneos sejam tratados de forma integrada e eficiente. As principais etapas de um projeto de *HW/SW co-design* são:

* **Especificação**

– Descrição dos requisitos do sistema com alto nível de abstração

– Linguagem de especificação deve permitir a descrição de requisitos funcionais e não-funcionais

– Exemplos de requisitos não-funcionais:

* + Desempenho, consumo de energia, custo, área, time-to-market, dependabilidade, ...

– Formalismos:

* + Autômatos, Redes de Petri, Máquinas de Estado Finito, Statecharts, Álgebra de processos, dentre outros.
* **Particionamento**

– Que componentes deverão ser implementados em hardware e quais em software

– De acordo com estimativas (métricas).

* + Necessidade de ter bons estimadores.
  + Estimadores fiéis e precisos

– Métricas:

* + Custo, tempo de execução, área de silício, taxa de comunicação, consumo de energia, número de pinos, área de memória, tamanho de programas, etc

– Métodos de particionamento:

* + hierarchical clustering, min-cut, simulated annealing, programação linear inteira e outros.
* **Co-síntese**

– Saída do particionamento: conjunto de módulos comunicantes (uns em hardware outros em software)

– Também chamado de protótipo virtual

– Co-síntese é o mapeamento (automático) do protótipo virtual para o protótipo real, de tal forma que todas as restrições são satisfeitas.

**–** Algumas decisões são tomadas nessa fase:

* + Escolha do processador, a rede de interconexão, protocolos de comunicação, interface entre hardware e software, escalonamento de processos concorrentes, dentre outros
* **Análise e Validação**

– A análise consiste em prover métricas de qualidade

– Validação pode ser feita após cada fase.

– Pode requerer interação entre diferentes ambientes de simulação (C e VHDL, p. ex.)

– Interação com um ASIC sem que este esteja implementado

**[QUESTÃO 05]. Defina Microprocessador e Microcontrolador.**

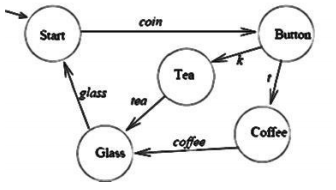
Microprocessador é um circuito integrado que contém milhares, ou mesmo milhões, de transistores.

O Microcontrolador é um microprocessador de propósito especial. Estas pastilhas contêm todos os circuitos integrados periféricos necessários aos equipamentos normalmente utilizados na área de controle de processos. Como o nome indica, são microprocessadores destinados a controlar.

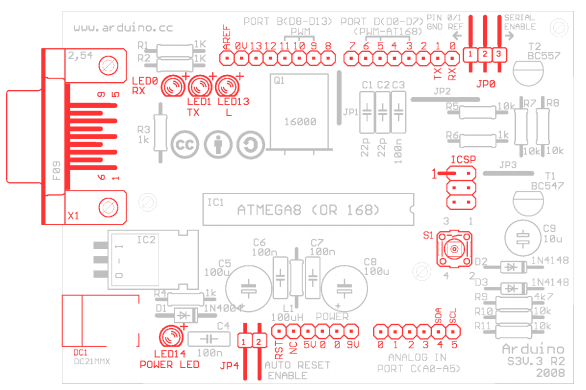
**[QUESTÃO 06]. Dado o fluxo de estados abaixo de uma máquina de café e chá. Implemente este projeto no micro-controlador 8051, apresentando:**

* Uma descrição da pinagem utilizada para o projeto
* A máquina de estados de Mealy usando a tabela de transição de estados e o diagrama
* O código da máquina de Mealy no micro-controlador 8051

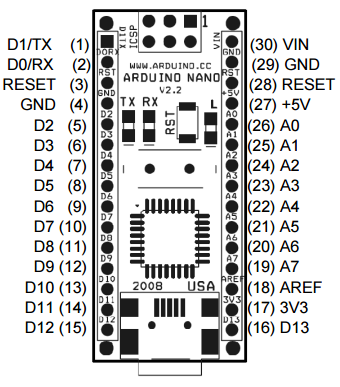
Descrever a sua solução com o máximo de detalhes possível, por exemplo, apresentando testes com simulador, inclusive descreva a forma como os testes foram feitos.



**[QUESTÃO 07]. Pesquise e selecione 2 micro-controladores ou plataformas com microprocessadores (exceto o 8051) e apresente o esquema de pinagem (onde está localizado e sua função) para programação. Adicionalmente, apresente 1 exemplo de código utilizando a pinagem da placa para cada escolha.**



O Severino é uma plataforma como um microcontrolador, neste caso o Atmega8 ou Atmega168 juntamente com um conjunto de pinos que possuem funções especificas. Os pinos JP0 são responsáveis pela conexão a plataforma, quando os pino 2-3 estão ativo, a conexão é feita pela porta serial e quando os pino 1-2 estão ativo, a conexão é feita dos pinos pin0(Tx) e pin1(Rx). Os pino JP4, quando 1-2 ativos executam o reset automático do microcontrolador. Ao lado do JP4 são os pino de energia Apresentando dois pino GND um pino de 9v e um de 5v. Ao lado do JP0 são os pino digitais de entrada e saída, onde de 0-7 são a porta D e de 8-13 as portas B. Ao lado dos pino de energia são os pino de saída analógica correspondentes aos pino de 0-5.



O Arduino Nano é composto por 30 pinos onde os pino de 1 e 2 e de 5 a 16 são os pinos digitais de entrada e saída, numerados de 1 a 13. Os pinos 3 e 28 são os pinos utilizado pra reset da placa. Os pinos 4 e 29 são os pino de 0v ou GND. Os pino 19 a 26 são os pinos de saída analógica. O pino 27 é o responsável pela voltagem, sendo o mesmo de 5V. O pino 30 é o pino de suporte a energia, sendo o pino de PWR. Os pinos 0 e 1 também podem ser utilizados para conexão com a placa.

**[QUESTÃO 08]. Descreva as funções da interrupção por Timers/Counters no micro-controlador 8051.**

O timer é um periférico do 8051 utilizado para gerar uma base de tempo bastante precisa. Toda vez que o timer conclui a sua temporização, o fluxo normal do programa é interrompido e então é executada uma função definida pelo programador, como por exemplo, incrementar um relógio. Este é um periférico acionado indiretamente pelo cristal ligado ao microcontrolador, só que esse clock é dividido por 12 antes de entrar nos Timers.

**[QUESTÃO 09]. Imagine um sistema de embalagem de produtos. O sistema controla esteiras, uma com produtos e outra com caixas de embalagem. No final da esteira de produtos, o produto é "derrubado" dentro de uma caixa que está na esteira logo abaixo. A caixa tem certa capacidade de produtos. Um sensor no final da esteira de produtos (um feixe de luz, por exemplo) detecta a queda do produto e envia um pulso a cada produto. Esse sinal deve ser conectado ao pino que mapeia a entrada do contador (timer/counter), o qual deve ser ajustado para a capacidade da caixa. Após o enchimento da caixa é gerada uma interrupção. O tratamento dessa interrupção deve então:**

* **Parar a esteira de produtos;**
* **Andar a esteira de caixas para posicionar uma nova caixa; e**
* **Acionar novamente o contador e a esteira de produtos. Fazer um programa para o micro-controlador 8051 de tal forma que o sistema fique livre para quaisquer outras tarefas enquanto a caixa não está cheia, ou seja, evitar o busy-waiting. Descrever a sua solução com o máximo de detalhes possível, inclusive a forma como os testes foram feitos. Fazer quaisquer suposições que se fizerem necessárias, por exemplo, mesmo não tendo o hardware do sensor, supor que o mesmo envia o pulso, o qual é lido por um pino do microcontrolador.**

#include<at89x52.h>

#define esteira\_caixa P0\_0

#define esteira\_produto P0\_1

int cont0, cont1, quantidade = 0;

void nova\_caixa(void) interrupt 3{

TR0 = 0;

TR1 = 0;

TH1 = 0x3C;

TL1 = 0xAF;

TR1 = 1;

cont1++;

if(cont1 == 60){

esteira\_produto = 0;

esteira\_caixa = 1;

quantidade = 0;

cont1 = 0;

}

}

void produto\_caixa(void) interrupt 1{

TR0 = 0;

TH0 = 0x3C;

TL0 = 0xAF;

TR0 = 1;

cont0++;

if(cont0==10){

esteira\_produto = 1;

esteira\_caixa = 0;

quantidade++;

cont0=0;

}

}

main(){

P0 = 0;

EA = 1;

ET0 = 1;

ET1 = 1;

TMOD = 1;

TR0 = 1;

while(1){

if(quantidade==10){

TR1 = 1;

}

}

}