

SLIDE 1 – INTRODUÇÃO

- O que caracteriza um sistema embarcado?
 - Inteligência embutida.
 - Tarefa específica.
 - O transformar em algo novo.
- Dilema entre Embarcado x Embutido(Inglês). O termo embutido é mais claro.

DEFININDO SEMB

- Conceito de Wolf: qualquer dispositivo que inclui um computador programável, mas que não é um computador de propósito geral. Tira vantagem de características da aplicação para otimizar o projeto.
- Um computador para agregar funcionalidades à um determinado dispositivo.
- OBS: A intel quando faz computador de propósito geral ela tem que rodar qualquer programa. Já quando faço para embarcado, posso ter restrições, mas em contrapartida, posso otimizar alguns pontos: preço, consumo de energia e desempenho. O preço do computador deve ser mínimo comparado ao preço do produto, não deve ocorrer muito impacto no preço.
- “Sistemas embarcados estão ficando mais complexos”
- “A computação não é um fim, mas um meio para alcançar o objetivo do usuário”
- “Computador não é o critério principal quando escolhemos um produto”
- “Equipamento existiria sem a computação”

CATEGORIAS

- Eletrônica de consumo
- Telecomunicações
- Veículos
- Controle industrial

CARACTERÍSTICAS (RESTRIÇÕES DE PROJETO)

- Funcionalidade sofisticada
 - Houve um crescimento na sofisticação capaz de ser realizada com sistemas embarcados, mas hoje a necessidade

de features sofisticadas, mas as restrições diminuem essa característica.

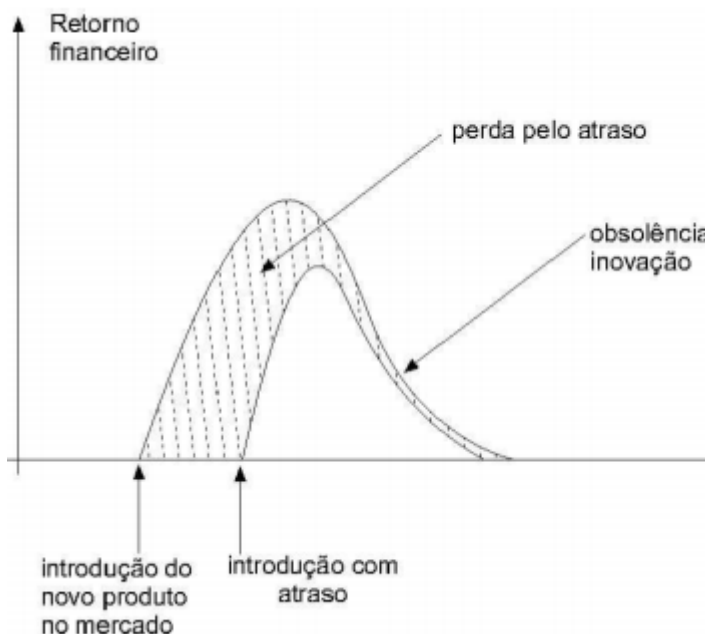
- Baixo consumo de manufatura *
 - Quanto mais escala melhor o resultado, vender um chip que vale um milhão para um milhão de pessoas dilui o custo do chip um milhão de vezes.
- Baixo consumo *
 - Tem que se ter cuidado com a bateria.
 - Potência está ligada ao aquecimento, quanto maior a potência maior o calor.
 - Energia está ligada à vida útil das baterias, e é a capacidade de um sistema de realizar suas funções.
 - Existe uma maior preocupação com a potência do que com a energia, já que a energia está ligada à potencia dissipada durante um tempo se melhorarmos a potencia estremos melhorando a energia,
- Desempenho *
 - Velocidade: Antigamente os sistemas embarcados realizavam tarefas consideradas “bobinhas”, hoje em dia existem aplicações mais “obscuras”, os processadores são melhores e as aplicações exigem mais processamento. e.g. Criptografia.
 - Deadlines: Existem aplicações embarcadas de tempo real na qual o resultado e tempo são importantes. O resultado deve ser entregue corretamente no tempo correto. Se o resultado não chegar no tempo certo ele não interessa mais. Existem aplicações de tempo real:
 - Soft, onde o desrespeito ao critério de tempo não causa danos sérios, por exemplo a renderização de um vídeo.
 - Hard, onde podem haver danos a vida das pessoas, por

exemplo em sistemas drive by wire.

- Equipe pequena e tempo curto.
- Interface com o usuário
 - Nem sempre é possível acoplar uma tela de touchscreen ao sistema, ou seja, existem restrições de como a comunicação com o usuário é feita.
 - Isso pode dar trabalho.
- Tamanho/ peso/ forma *

*requisitos não-funcionais -> São requisitos que não pode ser inclusos no código, o código ou processador não influenciam diretamente.

TIME-TO-MARKET



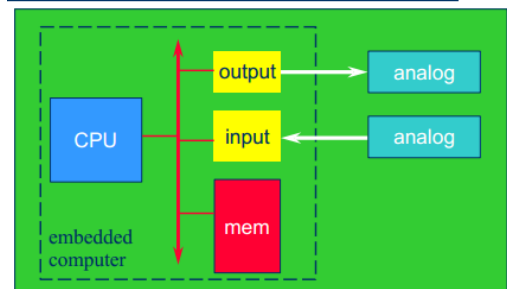
- Time-to-market: é o tempo desde a concepção do produto até o seu lançamento no mercado.
 - Ao introduzir um produto com atraso no mercado você perde o potencial retorno financeiro que ele daria.
 - A entrada de um concorrente também diminui o retorno financeiro do produto.
 - OBS: custo de desenvolvimento é 20% do custo total.
 - Se o produto demora muito tempo para entrar no mercado ele pode ser “passado para trás” por um

concorrente, a saída seria introduzir uma feature nova no produto para lança-lo.

- OBS: os engenheiros tem influencia no retorno financeiro do produto pois eles influenciam no time-to-market. O sucesso de vendas também depende dos engenheiros.
- TEMPO É O FATOR MAIS IMPORTANTE NO DESENVOLVIMENTO

COMPUTADOR EMBARCADO

- Um computador embarcado é um computador de Von Neuman com CPU, entrada e saída e memórias.
- A beleza da aplicação está na entrada/saída.



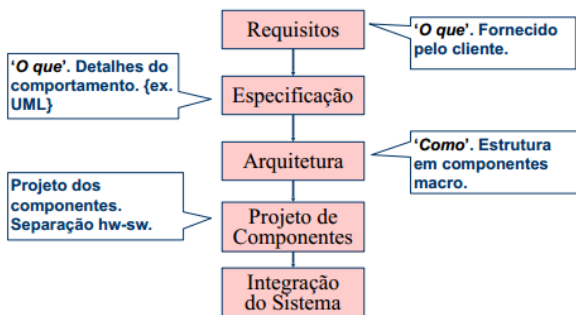
DESAFIOS DE PROJETO

- Quanto hw é necessário?
 - Quanto de CPU?
 - Quanto de memória?
- Como atender aos deadlines?
 - Mexer no hw ou sw?
- Como minimizar a energia?
 - Desligar lógica desnecessárias ou diminuir os acessos à memória?
- Funciona?
 - A especificação é correta?
 - A implementação atende a especificação?
 - Como testar as propriedades de tempo-real?
 - Como testar com dados reais?
- Qual plataforma de desenvolvimento?
 - Microprocessadores ou hw especializado? Ou ambos?

FAZEM PARTE DO PROJETO

- Escolher o hardware
 - CPU
 - Clock
- Deadlines
 - Melhorar HW e SW
- Energia
- Testes
- Plataforma

NÍVEIS DE ABSTRAÇÃO



SLIDE 2 – UNIDADE DE PROCESSAMENTO

- Formas diferentes de fazer computação.
- SW e processadores vão predominar.

FORMAS DE FAZER COMPUTAÇÃO

- ASIC – Application Specific Circuits
 - Específicos para uma aplicação
 - Pode ser toda ou parte de uma aplicação.
 - É algo feito sob demanda.
 - O custo é pago por um indivíduo/empresa apenas.
 - Prazo mais longo.
 - Custo altíssimo.
 - Eficiente no tempo e ineficiente no hardware.
- ASSP – Application Specific Standard Product
 - Variação do ASSIC.
 - Aplicação específica, mas vendido em prateleira.
 - Custo de desenvolvimento é dividido entre compradores.
 - Prazo imediato.

- Alto custo.
- “Preço sai maior”
- Processadores
 - Roda “qualquer algoritmo”
 - Comportamento definido “a posteriori”
 - Hardware do ASSIC não tem a mesma flexibilidade.
 - Eficiente no hardware e ineficiente no tempo.
- Reconfigurável
 - Hardware programável pelo programador
 - Parece um processador, mas não é.

Espaço de projeto; Suas opções para realizar um projeto, quanto mais experiência maior será seu espaço e mais opções você terá para explorar.

EFICIÊNCIA DE HARDWARE.

- Se tem pouco tempo, faz mais rápido, usa o hw num pulso.
- Se tem muito tempo faz mais lento para economizar energia.
- Quanto mais novo melhor a eficiência.
- Quanto mais específico melhor a eficiência.

PROJETANDO MIC

- Software
- Hardware
- Depuração
- Testes

PORQUE USAR MIC?

- Eficiente
 - Ponto de vista do custo, todos pagam pelo desenvolvimento do chip
 - Todos ganham.
 - MCU é simples e genérico.
 - Mercado alto para processadores.
 - Um MCU pode dar bom desempenho, se o hardware dedicado não for MUITO melhor, melhor ficar com o MCU.
 - Flexibilidade.

QUÃO GRANDE PRECISA SER A PLATAFORMA?

- Como escolher?
- Na nuvem X local?
- Quanto de memória?

PC COMO PLATAFORMA

- Antigamente placas não tinham escala
- Vantagens:
 - Barato e fácil de conseguir
 - Fácil para desenvolver.
- Desvantagens
 - Gasto de energia
 - Sobra muito hardware.
 - Não é muito confiável.
 - Não é adaptado para tempo-real.
- SW de um pC
 - Bios é um conjunto de softwares de entrada e saída.
 - Boot – Carrega um código no disco.
 - Nem toda plataforma usa SO como conhecemos.

PLATAFORMA

- Família de arquiteturas que satisfazem a um conjunto de restrições impostas para o reuso de componentes de hardware e software.
- Portar da família pic18f para pic24f é fácil.
- Plataformas promovem facilidade de reuso de hardware e software.
- Plataforma é abrangente “Microchip”
- Plataforma é específica “Pic18”
- Plataforma é um conjunto de coisas que é base para um produto.

DEPURANDO SEMB

- É difícil
- Existem ferramentas
- Host X Target
- Cross compiler, compilador transplataforma, compila em uma plataforma, mas gera código para outra.
- Cross-debugger, mostra estado do sistema alvo, permite controla-lo.

- Depurador in-circuit – depura o MC diretamente Permite interromper a execução e examinar os estados dos registradores.
- Emuladores/Simuladores erram mais.

Exercícios *

1) O que diferencia um sistema computacional embarcado de um sistema computacional ‘convencional’?

2) Defina time-to-market. Comente a sua importância em sistemas embarcados.

3) Defina SoC (System on Chip). Comente sobre as vantagens de usá-lo no projeto de um sistema embarcado.

4) Considere os seguintes sistemas embarcados: Um tablet, uma impressora e um sistema de navegação automotivo. Faça uma tabela com uma coluna para cada dispositivo e uma linha para cada uma das seguintes métricas de projeto: custo unitário, desempenho, tamanho e potência. Para cada campo da tabela, indique se existe uma forte restrição para aquela métrica.

5) Cite as estratégias utilizadas para implementar o processamento em computadores e destaque as diferenças entre elas.

6) Compare um microprocessador, que usa uma estrutura previamente definida e genérica com um circuito dedicado. Apresente pontos positivos e negativos de cada uma das duas estratégias.

5) Defina plataforma.

6) O que é um ‘Cross compiler’? Qual a sua importância no desenvolvimento de sistemas embarcados?

7) Ao se escolher um processador para ser usado em um projeto, em que o tamanho da palavra de dados (barramento) é relevante? E a palavra de instruções?

8) Explique porque um processador de propósito geral pode custar mais barato que um processador projetado por você mesmo. Por que, então, você faria seu próprio projeto?

* Os exercícios foram retirados do material disponibilizado pelo professor.