**1.Introdução**

A incorporação de sistemas eletrônicos numa vasta gama de produtos (como, por exemplo, automóveis, eletrodomésticos e aparelhos de telecomunicações) é o principal motivo pelo enorme crescimento da indústria eletrônica nos últimos anos. Já faz parte do senso comum que sistemas computacionais estão sendo amplamente utilizados, fato provado pelos milhões de sistemas produzidos anualmente para computadores de mesa (desktop), servidores, mainframes e etc. Porém, a surpresa é a escala em que sistemas com diferentes objetivos estão sendo fabricados todo ano (na ordem de bilhões), e se encontram embutidos nos mais variados equipamentos eletrônicos do nosso dia-a-dia. A utilização desses sistemas eletrônicos em aplicações corriqueiras têm resultado em produtos cada vez mais eficientes e mais baratos, apesar do aumento da complexidade dos respectivos projetos.

Definição:

Sistema embarcado é um sistema microprocessado, onde o conjunto é totalmente encapsulado e dedicado ao sistema que deve controlá-lo. É um sistema completo e independente, desenvolvido para realizar tarefas bem específicas e (grande parte das vezes) pré-definidas, cujo programa geralmente não pode ser acessado pelo usuário final, e que este poderá interagir através de interfaces, como displays, teclados, etc, desde que tenha sido projetado para tal finalidade.

Sistemas embarcados podem ser encontrados em:

* produtos de consumo: telefones celulares, pagers, câmeras digitais, aparelhos de DVD, vídeo games, calculadores, etc;
* eletrodomésticos: forno de microondas, fogão, geladeira, secretárias eletrônicas, equipamentos de segurança, máquinas de lavar, sistemas de iluminação, etc;
* automação de escritório: máquinas de fax, copiadoras, impressoras e scanners, etc;
* automóveis: controle de transmissão, injeção eletrônica, suspensão, freio ABS, etc;

Pode-se até mesmo afirmar que qualquer equipamento que utiliza eletricidade possui eu possuirá num futuro próximo sistemas de computação embutidos (sistemas embarcados). Vale ressaltar que o crescimento dos sistemas embarcados é superior ao dos sistemas computacionais tradicionais, na medida em que bilhõesde microprocessadores embarcados foram vendidos nos últimos anos, enquanto que e venda de microprocessadores para computadores pessoais é da ordem de milhões.

Logo, o mercado de sistemas embarcados tem se tornado cada vez mais atrativo e, ao mesmo tempo, mais rigoroso quanto aos aspectos de custo, tempo de desenvolvimento, qualidade do produto final, eficiência.

**2. Características Principais**

Enquanto um computador usual (desktop/notebook/PDA) serve aos mais diversos propósitos: checar seu orkut/email, usar seu editor de texto, usar a Internet, etc., um sistema embarcado é implementado para um propósito específico.

Em geral, um sistema embarcado:

* É um sistema construído para executar uma tarefa, completamente ou parcialmente independente de intervenção humana.
* É especialmente desenvolvido para concretizar pequenas ações da forma mais eficiente possível.
* Restrições de projeto mais rígidas: todos os sistemas de computação possuem em geral alguma restrição de projeto a ser garantida, como por exemplo, custo, tamanho, desempenho, potência dissipada, etc. Nos sistemas embarcados, no entanto, estas restrições são normalmente mais rígidas, incluindo várias restrições em conjunto.

* Sistemas reativos de tempo real: muitos sistemas embarcados devem reagir a mudanças no ambiente e devem fornecer resultados em tempo real. Exemplo: um piloto automático (cruise controller), que continuamente verifica a aceleração do carro, monitorando a velocidade e os sensores de freio do carro; sensor de temperatura.

Por definição, um sistema embarcado contém processador e software. Certamente, devido ao software, também deve haver memória para armazenar o código executável e os dados temporários gerados durante seu funcionamento. Ela pode ser do tipo ROM ou RAM; mas geralmente todo sistema embarcado possui ambas. Se apenas uma pequena quantidade de memória for necessária, ela estará contida no mesmo chip do processador. Caso contrário, as duas se encontrarão em chips de memória externos.



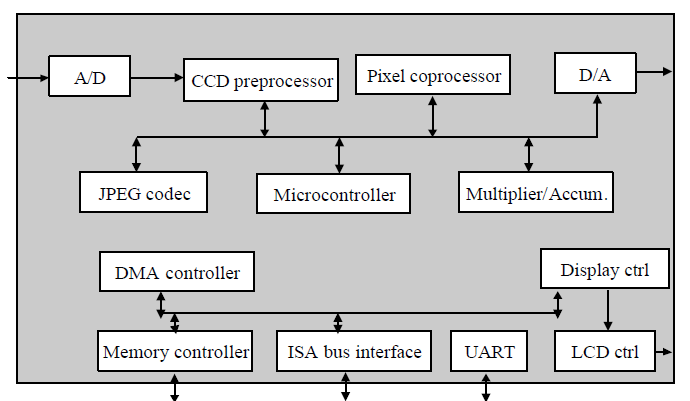
**Exemplo** **genérico de um sistema embarcado**

Uma grande parte dos sistemas embarcados são sistemas de tempo crítico ou tempo real (time critical or real-time applications), isto é, são sistemas que funcionam em ambientes nos quais o controle de tempo (timing) é muito importante: os resultados de determinada operação só são relevantes se ocorrerem num determinado tempo. Um sistema de controle de vôo de um avião é um sistema embarcado de tempo crítico: em caso de algum problema durante o vôo, se o dispositivo de controle não detectar o problema prontamente e solucioná-lo de forma adequada, em milissegundos, poderá haver conseqüências catastróficas.

É interessante notar que um computador de propósito geral é composto de uma série de sistemas embarcados: teclado, mouse, HD, modem, sound card, etc. Cada um desses sistemas possui um processador e software embutidos, bem como tarefas específicas: o modem, por exemplo, envia e recebe dados de forma digital através da linha de telefone analógica. Se um sistema embarcado for bem projetado, a existência de um processador e de software pode passar completamente despercebida por um usuário de tal dispositivo. Esse é o caso do forno micro-ondas, do DVD, do relógio digital, da cafeteira, da máquina de lavar, do controle remoto, entre outros. Em alguns casos, seria possível construir um dispositivo equivalente, que não contivesse um processador nem software. Isso poderia ser feito substituindo tal combinação por um circuito integrado que realizasse as mesmas tarefas em hardware puro. Entretanto, muita flexibilidade se perde, pois é muito mais fácil e mais barato mudar algumas linhas de código, do que projetar novamente uma peça de hardware. Além disso, de maneira geral, os sistemas embarcados não possuem muita flexibilidade (nem de software, nem de hardware).

A única flexibilidade permitida é o caso de upgrades para novas versões, fazendo com que o sistema possa ser reprogramado, com correções de bugs e implementação de novas funções emandadas pelo mercado (através do chip FPGA, por exemplo). Essas atualizações são, na maioria das vezes, feitas pelos fabricantes, e em casos particulares, pelos usuários finais.

Vamos considerar uma câmera digital para analisarmos alguns pontos importantes nos sistemas embarcados. Os circuitos A/D e D/A fazem a conversão de um sinal analógico para digital e vice-versa. O circuito JPEG codec realiza a compressão de descompressão de uma imagem no formato JPEG permitindo um armazenamento compacto de informações na reduzida memória da câmara. O coprocessador de pixels permite a rápida visualização das imagens. O controlador de memória e o DMA controlam o acesso a memória sem a necessidade de se utilizar o microprocessador. Os circuitos controladores de display e LCD controlam a visualização das imagens no display de cristal líquido. O circuito Multiplicador/Acumulador permite alguns tipos de processamento no sinal e o microcontrolador controla a ativação de todos os outros circuitos. Neste sistema cada módulo mencionado tem uma aplicação específica, enquanto que o microcontrolador é um circuito para aplicações mais genéricas.



**Esquema de uma câmara digital**

A maior parte dos sistemas embarcados inclui componentes de *hardware (*aplicação específica) e de *software (*componentes programáveis de propósito geral). Tenta-se quantificar sua qualidade medindo critérios específicos como desempenho, tamanho, custo de produção e de desenvolvimento, flexibilidade, peso, confiança. Tais critérios devem ser atingidos durante o projeto de hardware e de software. Assim, o desafio no projeto de sistemas embarcados é a obtenção de um resultado final que implemente a funcionalidade desejada e que, ao mesmo tempo, satisfaça à todas as restrições de projeto.

**3. História**

Nos primeiros anos dos computadores digitais na década de 1940, os computadores eram por vezes dedicados a uma única tarefa. Eram, entretanto, muito grandes e caros, comparados com os sistemas embarcados de hoje em dia. O termo sistema embarcado tem sua origem no fim da década de 1960. Nessa época o que existia era um pequeno programa de controle funcional de telefones, escrito em assembler, que era usado em outros dispositivos (não eram considerados um sistema embarcado em si).

Também na década de 60 surgiu o primeiro sistema embarcado reconhecido, durante o desenvolvimento do Projeto Apollo (conjunto de missões espaciais da Nasa para enviar o homem à Lua). Esse sistema era o Apollo Guidance Computer (usado para coletar e fornecer informações sobre voos, alem de controlar automaticamente todas as funções de navegação) , desenvolvido por Charles Stark Draper (1901 - 1987) no MIT. O primeiro sistema embarcado de produção em massa foi o computador guia do míssil nuclear LGM-30 Míssil Minuteman, lançado em 1961 pelos EUA. Ele possuía um disco rígido para a memória principal e era usado para controlar a orientação e estabilidade do foguete. Quando a segunda versão do míssil entrou em produção em 1966, o computador guia foi substituído por um novo, que constituiu o primeiro uso em grande volume de circuitos integrados (reduzindo o preço dos mesmos). A construção desse sistema imbarcado foi muito importante, pois ele estimulou as pesquisas com relação aos custos dos sistemas, diminuindo o preço dos mesmos.

Outro sistema embarcado utilizado em voos foi o sistema embarcado do caça Tomcat F-14 da marinha americana. Esse caça, lançado em 1970, continha o primeiro microprocessador baseado em sistema embarcado. O sistema era chamado de Garret CADC, controlava boa parte das funções do caça e tinha 8 processadores e 19 chips de memória. Na década de 1970 começavam a surgir bibliotecas de códigos direcionados para sistemas embarcados específicos com processadores específicos. Atualmente os sistemas embarcados podem ser programados em linguagens de alto nível e possuem sistemas operacionais, facilitando a sua programação.

Durante a década de 1970, os sistemas embarcados dó evoluíram graças a indústria de defesa militar. Todavia, devido a difusão dos sistemas embarcados, outras áreas estimulam o desenvolvimento desses sistemas. São elas: a indústria de jogos, a medicina e a aviação.

Desde suas primeiras aplicações, os sistemas embarcados vêm reduzindo seu preço e aumentando o seu poder de processamento e funcionalidade, principalmente após a década de 80. Nessa década, vários componentes externos foram integrados no mesmo chip do processador, o que resultou em circuitos integrados chamados microcontroladores. Esses microcontroladores não precisvam de ferramentas periféricas para executar o seu trabalho. Microcontroladores são mais baratos, porém menos flexíveis do que os tradicionais microprocessadores, mas mesmo assim essa ferramenta tecnológica possibilitou a difusão dos sistemas embarcados, estes atuando em areas que antes não se imaginaria um computador trabalhando.

Antigamente, microcontroladores de 16 e 32 bits tinham muitas desvantagens. O seu desenvolvimento era caro, os dispositivos eram grandes e pouco potentes. Por isso, era comum se utilizar microcontroladores de 8 bits apenas. Todavia, hoje em dia os preços caíram substancialmente e os dispositvos evoluíram. Assim, os microcontroladores de 32 bits passaram a ser utilizados, o que difundiu ainda mais os sistemas embarcados, pois estes puderam entrar em áreas que antes não eram aplicados, pois eram muito lentos.

Com o custo de microcontroladores menor que um dólar americano, tornou-se viável substituir componentes analógicos caros como potenciômetros e capacitores por eletrônica digital controlada por pequenos microcontroladores. No final da década de 1980, os sistemas embarcados já eram a norma ao invés da exceção em dispositivos eletrônicos.

Outro fator que ajudou na difusão dos sistemas embarcados foi a criação do Consórcio PC/104 pela Ampro, RTD e outros fabricantes. Esse grupo estabeleceu estabeleceu um formato para microprocessadores Intel baseado em uma placa-mãe de aproximadamente quatro polegadas quadradas, e um pouco menos de uma polegada de altura. As tábuas foram empilhadas, permitindo que um computador muito potente pudesse ser montado em uma caixa de aproximadamente quatro polegadas quadradas. O PC/104 foi inicialmente dirigido aos militares e às áreas relacionadas à saúde, onde tornou-se amplamente utilizado. Quando o poder do processador cresceu suficientemente lidar com aplicações multimídia, o PC/104 passou a ser utilizado em outras áreas.

[](http://pt.wikipedia.org/wiki/Ficheiro:Apollo_program_insignia.jpg)

**Insignia do projeto Apollo, que desenvolveu o primeiros sistema embarcado**

[](http://en.wikipedia.org/wiki/File:Charles_Stark_Drape.jpg)

**Charles Stark Draper, criador do primeiro sistema embarcado.**

[](http://en.wikipedia.org/wiki/File:Dsky.jpg)

**Interface do Apollo Guidance Computer**



**O caça F-14, controlado por um sistema embarcado.**

**4. Tecnologia Empregada**

Os projetos de sistemas embarcados apresentam uma grande flexibilidade não só do ponto de vista do software, mas também do hardware. Sua interação com o ambiente normalmente requer o uso de sensores e outros dispositivos que normalmente não estão presentes em sistemas de propósito geral.

* Hardware

**Dispositivos Processadores**

Processadores são dispositivos usados para transformar ou mover dados e/ou tomar decisões sobre ações a serem executadas. Assim, vários dispositivos podem ser considerados processadores. Alguns seguem a linha mais tradicional, sendo programados via software e, portanto, apresentando grande flexibilidade para alterações de sua funcionalidade e comportamento, os chamados *processadores de software.*

Entretanto, podemos ter dispositivos de hardware desenvolvidos especificamente para desempenhar uma determinada função. Estes dispositivos são extremamente rápidos, mas não são flexíveis e caso seja necessária alguma alteração de seu comportamento, deverão ser substituídos por outro dispositivo, chamados de *processadores de hardware.*

É importante fazer uma distinção entre o processador que está sendo usado para desenvolver o projeto e aquele(s) onde o sistema embarcado será implementado. O primeiro é chamado de *processador de desenvolvimento*, onde temos o ambiente de desenvolvimento, como editores de texto, compiladores, depuradores, etc, e o segundo é o *processador alvo*.

**Processadores de propósito geral**

Processador de propósito geral é um processador projetado para uma grande variedade de tarefas computacionais. Apesar da possibilidade de utilização de processadores de propósito geral (Pentiums, PowerPCs) para a implementação de sistemas embarcados, normalmente não são escolhidos, há que seus fabricantes deixam muitas tarefas para serem implementadas externamente, por componentes auxiliares. Isto porque busca-se tornar os projetos de computadores mais flexíveis. Esta flexibilidade tem vantagens do ponto de vista de desempenho, porém adiciona um custo extra que pode

inviabilizar o projeto do sistema embarcado.

Uma solução alternativa encontrada pelos fabricantes é criar “processadores embarcados” (*embedded processors*) baseados em processadores de uso geral que incorporam vários dispositivos que facilitam e barateiam o projeto de sistemas embarcados. A grande vantagem em utilizar este tipo de processador está na fase de desenvolvimento do projeto. Pode-se usar um desktop que utilize um processador da mesma família do processador que se deseja usar no sistema embarcado. Neste caso, o processador de desenvolvimento e o processador alvo são o mesmo ou, pelo menos, pertencem à mesma família. Assim, ferramentas como compiladores, depuradores e ambientes completos de desenvolvimento para softwares de desktop podem ser utilizados no desenvolvimento do sistema dedicado com facilidades de visualização dos testes na tela do desktop. Para verificar a interação do sistema com os dispositivos externos de entrada e saída, o projetista pode optar por instalar estes dispositivos diretamente no desktop ou simular seu comportamento por meio de software.

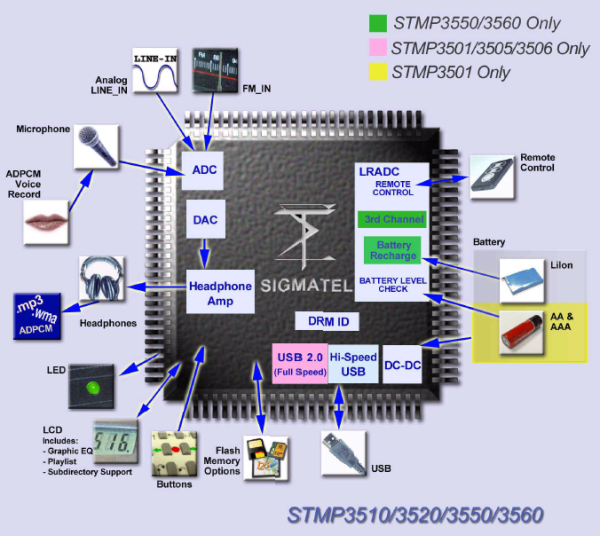
Por serem bastante utilizados, e geralmente não necessitarem de processadores muito robustos, os sistemas embarcados fizeram com que o processador Z80 seja o processador mais produzido até hoje. Esse processador, de 8 bits, é muito simples, e o que torna tão usado é o fato de ele ser extremamente barato e econômico. O Z80 é comumente utilizado em mp3 players. Outro exemplo de processador muito utilizado é o Motorola 68000, que possui 32 bits.

Para sistemas que necessitem de um maior poder de processamento, existem as famílias de processadores ARM, que utilizam a arquitetura RISC (computador com conjunto mínimo de instruções, em inglês), que são mais eficientes, rápidas e possuem um bom desempenho. Esse tipo de processador, juntamente com alguns MB de memória, consegue rodar o Linux ou o Windows Mobile. Os chips ARM são baratos e possuem um baixo consumo elétrico, por isso são extremamente populares em celulares, PDAs, modems ADSL, videogames e assim por diante. Cerca de 75% de todos os processadores de 32 bits usados em sistemas embarcados são processadores ARM.Um exemplo de sistema embarcado é esse MP4 player retratado abaixo. Ele possui apenas três chips: um chip de memória flash, um controlador principal e um sintonizador de radio AM/FM.



MP4 player

O chip principal, um Sigmatel STMP3510, é um micro-controlador que desempenha sozinho todas as funções do aparelho, controlando as funções disponíveis e possui até uma pequena quantidade de memória RAM:



Funções controladas pelos chips Sigmatel STMP

É este tipo de micro-controlador que permite que modems, MP3 players, celulares e outros aparelhos tenham seu preço final diminuído ao longo dos anos, pois com menos chips, o custo cai proporcionalmente. Este tipo de chip, se comprado em larga escala, chega a custar apenas 6 dólares.

Para aplicações onde um chip personalizado é essencial, existe a opção de usar FPGA’s ou de LCAs (logic-cell arrays). Esses chips são compostos por um enorme número de chaves programáveis, que podem ser configurados para simular o comportamento de qualquer outro circuito. Como já foi dito, um único FPGA pode simular não apenas um processador simples, mas também outros circuitos de apoio, como o controlador de vídeo ou interface serial. Os FPGAs são muito mais caros que os chips produzidos em larga escala, mas são uma boa opção quando necessita-se de apenas algumas centenas de unidades de um design exclusivo.

**Microcontroladores**

Microcontroladores são processadores de software com uma filosofia muito semelhante à dos processadores embarcados mencionados anteriormente, ou seja, incorporam muitas funções num único chip. Não há um consenso quanto a distinção entre microcontroladores e processadores embarcados, contanto, consideraremos que os microcontroladores (diferente dos processadores anteriores) não são derivados de famílias de processadores de propósito geral usados em desktops e tem, geralmente, um poder de processamento menor.

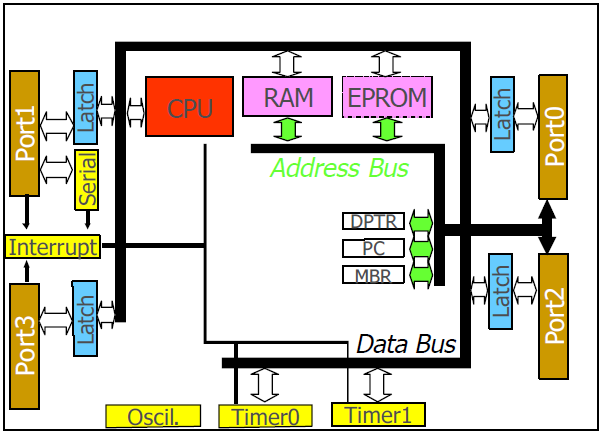
Como os microcontroladores são projetados especificamente para sistemas embarcados, é comum que apresentem um conjunto de instruções bem adaptado para este fim, como por exemplo, instruções de manipulação de bits ou acesso a pinos específicos do processador para facilitar a implementação de interfaces com dispositivos externos.

Um microcontrolador pode incorporar uma grande variedade de dispositivos como: conversores analógico-digitais (ADC) e digital-analógicos (DAC), temporizadores, contadores, interfaces seriais, memória de instruções e/ou dados, controladores de interrupção e etc. Por isso, é comum que não seja desenvolvido apenas um, e sim uma família de microcontroladores, cada um apresentando um conjunto diferente de dispositivos, freqüência de clock, potência consumida, faixa de temperatura suportada, encapsulamento e preços compatíveis com o mercado. Desse modo o projetista pode escolher o modelo que melhor se adapte aos seus requisitos técnicos e de custo. Além disso, quanto maior a família do microcontrolador mais vida útil o projeto terá, visto que é mais provável encontrar processadores da mesma família que incorporem mudanças futuras do projeto.

**Família 8051**

Esta é, atualmente, a família mais conhecida de microcontroladores. Inicialmente, lançada pela Intel, é atualmente fabricada por várias companhias como Philips, Atmel, Dallas Semiconductors, entre outros, o que garante preços baixos e uma enorme variedade de opções. Estes processadores apresentam, de um modo geral, baixo desempenho e consumo, embora já existam membros da família com performance consideravelmente boa.

Port1 Port3

Port2  **Processador 8051 típico**

ort0

**Outras famílias**

Existem várias outras famílias de microcontroladores. A escolha da família a adotar no projeto de sistema embarcado vai depender não apenas dos fatores técnicos mais comuns, como velocidade, potência, tamanho, mas também da facilidade de uso, ambientes de desenvolvimento existentes, conhecimento prévio do time de desenvolvimento, facilidade de compra, número de fornecedores, etc. Dentre as várias famílias podemos citar: ARM da Intel, PIC da Microchip, Série HC da Motorola e Transputers da SGSThomson.

**ASIPs**

Application-Specific Instruction-Set Processors são processadores especialmente desenvolvidos para uma determinada função, seja de controle, processamento de sinais, comunicação, etc. Seu conjunto de instruções e os periféricos incorporados são escolhidos de modo a se obter a melhor relação custo-benefício para o projeto em questão.

De um modo geral, o ASIP e seu compilador são projetados em paralelo de modo a garantir a melhor escolha de implementação de instruções, seja em hardware ou em software. Atualmente existem muitas pesquisas em andamento para permitir o desenvolvimento rápido de ASIPs e seus compiladores simultaneamente. De forma informal, podemos dizer ASIPs estão no meio do caminho entre processadores de uso geral e dispositivos de hardware especialmente desenvolvidos para uma determinada aplicação. Eles apresentam melhor performance que os primeiros mas menor que os segundos. Em contrapartida são mais flexíveis a mudanças que os segundos, já que sua programação é feita por software.

**Hardware Específico**

A opção mais radical com relação ao projeto de dispositivos específicos para uma dada função é construir um hardware dedicado à execução de um determinado algoritmo ou ASIC (Application- Specific Integrated Circuit). A opção por usar tais dispositivos só se justifica quando os requisitos de performance, consumo, tamanho, preço, etc não são possíveis de se obter por soluções de software executados sobre processadores de propósito geral. Isso se deve ao elevado preço para a fabricação de dispositivos sob encomenda e à perda de flexibilidade com relação a mudanças futuras do projeto que se façam necessárias. Uma opção menos radical está no uso de FPGAs que podem ser programados para executar qualquer tipo de algoritmo.

**System-On-A-Chip**

Para construir um sistema embarcado, é importante permitir o encapsulamento de todo o projeto num único chip, mesmo que o projeto seja composto por uma mistura de processadores de software e de hardware. Por exemplo, vamos supor que o projeto seja composto por um processador 8051 e um ASIC. Para implementá-los num único chip, precisa-se dispor de núcleo de processador, que corresponde a uma biblioteca contendo todo o projeto de um 8051 que pode ser incorporada ao chip em desenvolvimento por meio de ferramentas de CAD (*Computer-aided design* ou projeto assistido por computador).

As grandes vantagens em fazer implementações do tipo *system-on-a-chip são: o* custo da fabricação em série, a qualidade, a diminuição de defeitos de montagem e fabricação em geral, a potência consumida, o tamanho e a velocidade. Entretanto, este tipo de implementação só se justifica para grandes volumes de fabricação ou requisitos técnicos com muitas restrições.

**Memória**

Nos sistemas embarcados os tipos de memória utilizados podem diferir bastante dos tipos utilizados em computadores comuns. Discos rígidos não são utilizados normalmente (nem outra forma de armazenamento externo). Além disso, dependendo da aplicação, o tamanho da memória pode ser muito pequeno (como em aplicações de controle), e esses sistemas são projetados para trabalhar ininterruptamente sem falhas ou responder o mais rápido possível (ex: sistema de frenagem ABS).

A maioria dos desenvolvedores de software pensam que os únicos tipos de memória são (random access) RAM ou read-only (ROM). Num dispositivo RAM, os dados armazenados em cada localização de memória podem ser lidos e modificados a qualquer momento. Num dispositivo ROM, os dados armazenados podem ser lidos mas não modificados. Em alguns casos é possível sobreescrever os dados armazenados num dispositivo parecido com ROM. Tais dispositivos são chamados memórias híbridas porque elas exibem algumas das características de RAM e ROM.

Assim, o tipo de memória mais utilizado são as memórias não voláteis, que retém as informações após o desligamento do sistema. As memórias não voláteis normalmente usadas são do tipo EEPROM (Electrically-Erasable Programable Read-Only Memory), cujos bytes podem ser apagados independentemente (esse tipo de memória é mais flexível, porém mais lenta), ou Flash EPROMs, cujos dados são apagados de uma vez (menos flexível, porém mais rápida). As memórias voláteis utilizadas são normalmente do tipo SRAM (Static Random-Access Memory), alimentadas por baterias de modo a manter os dados em caso de falta de energia. Essas baterias podem manter os dados por até dez anos.

Um tipo de memória que atualmente está sendo bastante utilizado em sistemas embarcados é a memória flash, por conta da queda no seu preço. O problema desse tipo de memória é que ele funciona apenas como unidade de armazenamento e não dá para ser utilizada como memória de trabalho, tal qual os HDs nos computadores convencionais, necessitando de uma quantidade de memória RAM, que pode estar embutida no micro-controlador ou num chip separado.



**Tipos de memórias para Sistemas Embarcados**

**HDL**

Linguagem de descrição de Hardware (HDL) usa código para representar funções digitais. "Firmware" frequentemente se refere ao código HDL resultante. HDLs são uma abordagem comum e popular ao design de FPGA. Pode ser criado o código-fonte com qualquer editor de texto. Editores especiais de HDL como CodeWright e Scriptum oferecem recursos como “HDL templates” e destaca palavras reservadas. HDLs podem ser genéricos (suportados por muitas ferramentas de simulação e desenvolvimento) como Verilog ou VHDL (Very High Speed IC HDL), ou privados como Altera's Hardware Description Language (AHDL), que só e reconhecido pelo ambiente de desenvolvimento da Altera.

Há dois estilos de escrita para designs HDL, estrutural ou comportamental. Firmware estrutural usa componentes privados (“vendor-specific”) para construir as desejadas funções digitais, e por isso repetir o processo de design é necessário para fabricantes diferentes.

Firmware de HDL comportamental descreve funções digitais de forma genérica ou abstrata que são geralmente independentes de fabricantes. Isso fornece flexibilidade suficiente para reuso de código em FPGA’s de diferentes vendedores, com pouca ou nenhuma modificação de código. Vantagens do design comportamental são: sua flexibilidade e a economia de tempo e custos financeiros que ele propicia, e é pouco dependente de fabricante.

VHDL e Verilog são as linguagens HDL mais populares. Arquivos VHDL consistem de três partes principais: declaração de biblioteca, declaração de entidade, e seção de arquitetura. Mesmo não sendo requerido pelo VHDL um cabeçalho adicional pode ser incluído. Tal seção deveria conter informação pertinente, como o nome do desenvolvedor, nome do arquivo, um breve resumo sobre as funcionalidades do arquivo. Devido a similaridade entre HDL e software, firmware designers devem seguir regras usuais ao desenvolvimento de software.

**Interfaces externas**

Todos os sistemas embarcados devem interagir com o sistema no qual estão embutidos, podendo ainda interagir com o ambiente ou com outros sistemas. Essa interação é feita através de diferentes tipos de interfaces.

Um dos tipos de interfaces são as interfaces seriais, que são aquelas em que os dados são enviados bit a bit. A interface mais comum desse tipo é o padrão RS232C (*Recommended Standard-232*) da *Electronic Industries Association* (EIA) que suporta apenas comunicações entre dois dispositivos, e é bastante usado por modems, mouses e algumas impressoras. Na prática pode-se transmitir dados até cerca de 200kbps e os dispositivos podem ficar até 30 metros de distância.

A interface serial mais conhecida pode ser a Universal Serial Bus (USB) que começou a ser usado em 1998 e suporta taxas de até 12Mbps, além de poder conectar até 127 periféricos numa única linha. Os dispositivos USB podem ser conectados durante a própria execução, e são reconhecidos rapidamente pelo sistema.hospedeiro. O infravermelho também é utilizado como meio de comunicação em vários sistemas embarcados, como videocassetes, televisores, aparelhos de som, celulares, etc. O Bluetooth também é um meio de comunicação que ganhou muito reconhecimento nos últimos anos. Esse sistema utiliza ondas de rádio para transmissão de dados à curta distância. O Bluetooth é bastante versátil, pois pode transmitir dados através de objetos não metálicos além disso tem fácil integração com protocolo TCP/IP.

Outra interface bastante utilizada são os teclados e visores de cristal líquido que só apresentam letras, números e caracteres de pontuação. Alguns tipos de dispositivos são capazes de transformar um tipo de energia em outra. Esses dispositivos são os transdutores, que podem interpretar dados de maneira contínua (analógica). Por isso, para que o processador possa interagir com esses dispositivos, é necessário transformar o analógico em digital, ou seja, em números binários, que possam ser interpretados pelo sistema embarcado. De modo análogo, uma conversão digital-analógico deve ser feita no caminho contrário, para que os sinais digitais possam ser tratados pelo transdutor.

**Sistemas de Tempo Real e Tolerância a Falhas**

Como já dito, a maioria dos sistemas embarcados devem prover uma resposta num determinado intervalo de tempo. Sistemas cujo tempo de resposta é importante são denominados sistemas de tempo real.

Além disso, os sistemas embarcados devem garantir uma grande confiabilidade no seu funcionamento, principalmente após a sua difusão no mundo atual. Dessa forma, há um mecanismo chamado de tolerância de falhas, que visa lidar com potenciais problemas que possam afetar os sistemas embarcados. Esse mecanismo consiste em admitir que as falhas vão ocorrer, e também em achar um modo manter o sistema funcionando mesmo com as falhas ocorrendo.

Para se adquirir tolerância a falhas, faz-se necessário o uso duplicatas de cada ferramenta, tanto de software como de hardware. Porém, em sistemas embarcados há um problema: o volume, o peso e o consumo de energia do sistema são de extrema importância, logo a aplicação de tolerância de falhas deve ser bem dosada.

**FPGA**

FPGAs são chips digitais programáveis, ou seja, eles podem ser programados para realizar qualquer função digital, pois são compostos de um enorme número de chaves programáveis, que podem ser configuradas para simular o comportamento de qualquer outro circuito.

Internamente os FPGAs são compostos de blocos básicos, que por sua vez são compostos de uma lógica combinacional que pode serprogramada para implementar qualquer função booleana de 4 ou 5 variáveis, conforme o modelo usado. Cada bloco básico dispõe também de elementos de memória (flip-flops) que podemarmazenar os resultados obtidos pela função booleana. A conexão entre blocos básicos também é configurável, formando complexas estruturas combinacionais com armazenamento de estado nos blocos configurados como memória. Em resumo, é possível configurar o FPGA para executar qualquer tipo de algoritmo.

Após criar o programa desejado e convertê-lo em um arquivo binário com um compilador, podemos transferi-lo para o chip FPGA. Vale ressaltar que esse processo pode ser repetido várias vezes, logo o chip pode desempenhar diferentes funções dependendo da conveniência do usuário. FPGAs “perde” o programa quando é o fornecimento de energia é interrompido (como a memória RAM). Desta forma, para recuperar a antiga função, é necessário que o chip FPGA seja reprogramado.

Um único FPGA pode simular não apenas um processador simples, mas também outros circuitos de apoio, como o controlador de vídeo, uma interface serial e assim por diante. Existem vários tipos de FPGA, alguns com programação com tecnologia PROM, que só podem ser programados uma única vez, outros com EPROM, que podem ser reprogramados em laboratório, e ainda outros em RAM, onde é possível até reconfigurar durante a execução. Isso dá uma grande flexibilidade ao projetista aliado a uma performance em geral bem maior que a obtida com o uso de processadores de propósito geral.



*Chip FPGA*

* **Exemplo de Aplicação:**

O crescente avanço das *Field Programmable Gate Arrays* (FPGAs) tornou possível a criação e utilização de *soft-cores*, processadores projetados e implementados em linguagem de descrição de hardware. A flexibilidade das FPGAs permite que o projetista do processador possa adaptá-lo aos requisitos da aplicação.

**SISTEMAS EMBARCADOS: ACOPLAMENTO DO SOFT-CORE PLASMA AO**

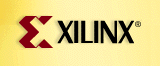
**BARRAMENTO OPB DE UM POWERPC 405**

*Rafael Vargas1, Rafael Cancian2, Hugo Marcondes3, Antônio A. Fröhlich4.*

Universidade Federal de Santa Catarina

Florianópolis, SC, Brasil

* Maiores produtores mundiais: Existem (no mínimo) 5 companhias produzindo FPGAs no mundo. As duas maiores são a XILINX e ALTERA.

[](http://www.xilinx.com/) [http://www.fpga4fun.com/images/Altera_logo.gif](http://www.altera.com/)

* Software

Programar sistemas embarcados não é como programar um PC comum. De fato programação para um sistema embarcado se assemelha a programação de computadores de 15 anos atrás. O hardware escolhido para o sistema é em geral escolhido para que o dispositivo seja o mais barato possível. Bem, isso significa que programador terá que lidar com processadores lentos e baixa memória, enquanto que ao mesmo tempo se esforçará para conseguir uma eficiência que não é vista na maioria dos aplicativos para PC.

Há um grande número de arquiteturas diferentes de sistemas embarcados e isso torna as ferramentas para programar mais caras. Além disso, essas ferramentas têm menos recursos e são menos desenvolvidas. Num projeto embarcado de grande porte, fatalmente você irá encontrar algum ‘’bug’’no compilador. Ferramentas de “debug” são outro problema, pois nem sempre você consegue rodar um “debugger” no seu processador embarcado devido à escassez de recursos e isso torna o controle de erros do programa difícil. Hardwares especiais como portas JTAG podem superar esse problema parcialmente. Debugging é de suma importância, pois se o sistema “travar” durante o funcionamento de um sistema de hardware real (como um motor, por exemplo), podem ocorrer danos permanentes ao equipamento.

Para um programador de sistemas embarcados, fazer uso dos algoritmos mais eficientes é essencial, bem como um entendimento da arquitetura de hardware sobre a qual você está trabalhando, de forma a facilitar a otimização do seu software.

Memória também é um problema, por ser escassa: diferentemente de programas para PC, o programador deve usar algoritmos que sacrifiquem tempo do processador em favor da memória. Ou seja, nada de “memory leak” (consumo de memória, não intencional, quando um programa falha em liberar memória que não é mais necessária). Frequentemente, aplicativos embarcados usam técnicas determinísticas de memória e evitam o uso de funções “malloc” (função C de alocação dinâmica), facilitando a investigação e solução de possíveis “leaks”.

**C e Assembly**

Muitos programadores preferem programar em C a programar em assembly, por boas razões, já que C é de alto nível e portanto libera o profissional do conhecimento e controle de alguns detalhes de implementação a nível de máquina.

Contudo há alguns processos de baixo-nível que ou podem ser melhor implementados em assembly ou **somente** podem ser implementados em linguagem assembly. Por outro lado é freqüentemente útil para o programador analisar o código assembly produzido pelo compilador C e manualmente editá-lo a fim de otimizá-lo de formas que o compilador não consegue. Assembly também é útil para processos de tempo crítico (time-critical) porque diferentemente do que ocorre com linguagens alto-nível não há ambigüidade quanto a como o código será compilado. O “timing” pode ser controlado de forma mais eficiente, o que é útil para produzir “drivers” simples.

A maioria dos programadores de C são “mal-acostumados” porque eles programam em ambientes nos quais há uma série de bibliotecas prontas para uso. O fato cruel é que em sistemas embarcados raramente há tantas bibliotecas às quais o programador habituou-se a usar, ocasionalmente um sistema embarcado pode nem ter uma biblioteca padrão completa, se sequer houver uma. Frequentemente, devido a problemas de espaço, não é possível incluir uma biblioteca inteira, e programadores são muitas vezes forçados a implementar suas próprias bibliotecas. Enquanto algumas bibliotecas são muito grandes e não muito adequadas ao uso em microcontraladores, muitos sistemas de desenvolvimento incluem as bibliotecas padrão mais comuns para programadores C.

C continua a ser uma linguagem muito popular para microcontroladores devido à sua eficiência de código e reduzido tempo de desenvolvimento. C oferece controle de baixo nível e é mais humanamente legível que assembly. Muitos compiladores C estão disponíveis para uma grande gama de plataformas de desenvolvimento. Os compiladores são parte de IDEs com uma janela de assembly. Adicionalmente, usando C há um aumento na portabilidade, uma vez que o código em C pode ser compilado por diferentes tipos de processadores.

## Bootloader

A fim de simplificar muitas tarefas, programadores de muitos sistemas usam um tipo genérico de software chamado **bootloader** que é responsável por realizar algumas rotinas do sistema (como: liberar o modo protegido), e então carregar o kernel na memória, para a partir daí transferir o controle do sistema para o kernel. Bootloaders são usados em muitos microcontroladores e são em geral a forma mais rápida para atualizar um programa, com pequenas mudanças, em um microcontrolador. Isso torna o ciclo editar-compilar-baixar-testar um pouco mais rápido.

**Ferramentas de Desenvolvimento:**

**O proceso de construção**

Existem muitas coisas que as ferramentas de desenvolvimento de software podem fazer automaticamente quando a plataforma alvo está bem definida. Essa automação é possível porque as ferramentas podem explorar recursos do hardware e do sistema operacional nos quais o programa será executado. Dessa forma o compilador, esconde do programador certos aspectos do processo de construção do software.

O processo de conversão do código-fonte para do software embarcado em um arquivo binário executável envolve três passos distintos. Primeiro, cada arquivo-fonte precisa ser compilado ou montado em um programa-objeto. Depois, todos os os programas-objetos originados a partir do primeiro passo necessitam ser interconectados para produzir um objeto único, chamado programa-relocável (“ relocatable program”). Por último, o endereço físico da memória devem receber os “offsets relatives” do programa-relocável no processo chamado realocação. O resultado do terceiro passo é um arquivo que contém um um programa-executável em binário que está pronto para ser rodado no sistema embarcado.



Figura: Processo de conversão do código-fonte para um arquivo binário do Sistema Embarcado

Cada um dos passos no processo de construção do software embarcado é uma transfromação realizada por software rodando num computador de propósito geral. Para distinguir esse computador, onde se realiza o desenvolvimento ( PC ou Unix workstation), do sistema embarcado alvo (target), ele é referido como o computador host. Em outras palavras, o compilador, assembler, linker são softwares rodando no host, e não no sistema alvo. Ainda assim, apesar do fato de rodarem em outra plataforma computacional, essas ferramentas se combinam para produzir um executável binário que irá rodar apropriadamente no sistema embarcado alvo.

****

**Divisão entre host e alvo**

Vale ressaltar o fato de os kernel’s monolíticos vem se tornando cada vez mais populares por seus vários benefícios. É um kernel relativamente grande com capacidades sofisticadas é adaptado para um ambiente embarcado. Isso dá ao programador um ambiente similar ao de um sistema operacional desktop como o [Linux](http://en.wikipedia.org/wiki/Linux) ou o [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), e, portanto o desenvolvimento é muito produtivo, por outro lado, tal kernel requer consideravelmente mais recursos de hardware, e por causa da sua complexidade esse tipo de kernel é menos previsível e menos seguro/confiável (maior chance de erros). Alguns exemplos de kernel’s monolíticos embarcados são [Embedded Linux](http://en.wikipedia.org/wiki/Embedded_Linux) e o [Windows CE](http://en.wikipedia.org/wiki/Windows_CE). Apesar de aumentar o custo do hardware, esse tipo de sistema embarcado está cada vez mais popular, especialmente em dispositivos mais poderosos como [Wireless Routers](http://en.wikipedia.org/wiki/Routers) e [GPS Navigation Systems](http://en.wikipedia.org/wiki/Automotive_navigation_system).

Há diferentes tipos de arquitetura de software em uso:

**SUPER LOOP:**

Um superloop é uma estrutura de programa composto de um loop infinito, com todas as tarefas do sistema contidas nesse loop. Aqui está um pseudocódigo de um superloop.

Function Main\_Function()

{

Initialization();

Do\_Forever

{

Check\_Status();

Do\_Calculations();

Output\_Response();

}

}

Nós colocamos as rotinas de inicialização antes do superloop, porque nós somente queremos inicializar o sistema uma única vez. Assim que o loop infinito seja iniciado, nós não queremos “resetar” os valores, porque precisamos manter estado persistente no sistema embarcado. Dessa forma, torna-se imprescindível o uso do superloop nos sistemas embarcados. Sistemas embarcados não são os únicos a utilizar tal arquitetura, jogos, por exemplo, constantemente usam um loop similar, o chamado *(tight) (main) game loop*.

## Power Save Superloop:

Digamos que tenhamos um sistema embarcado que tem um tempo de loop de 1ms, e só precisa checar certa entrada de dados uma vez por segundo. É um evidente gasto continuar a repetir o loop ininterruptamente, especialmente se não precisamos fazer nada na maior parte do tempo. Nessa situação, por exemplo, o programa irá executar 1000 loops antes de ler a entrada, e os outros 999 loops terão sido apenas uma contagem regressiva para a próxima leitura. Nesse caso, é muito ineficiente ter o processador rodando a 100% todo o tempo. Seria muito melhor implementa um atraso (delay) no programa, tornando esse sistema embarcado mais eficiente. O superloop extendido seria:

Function Main\_Function()

{

Initialization();

Do\_Forever

{

Check\_Status();

Do\_Calculations();

Output\_Response();

**Delay\_For\_Next\_Loop();**

}

}

Se tal atraso for de 999ms, não precisamos de 1000 loops e podemos ler a entrada em cada loop.

Desenvolvedores de sistemas embarcados usam [compiladores](http://en.wikipedia.org/wiki/Compiler), [assemblers](http://en.wikipedia.org/wiki/Assembly_language#Assembler) e [debuggers](http://en.wikipedia.org/wiki/Debugger) para desenvolver softwares de sistemas embarcados. Contudo, eles também usam ferramentas mais específicas. Entre elas estão ferramentas , matematicas como o MATLAB (software voltado para o calculo numerico), ou ainda se pode personalizar uma linguagem de programação para otimizar o seu uso (Java, Pascal).

DEBUGGING

Como um sistema embarcado é geralmente composto por uma série de elementos, a estratégia de debugging pode variar. Por exemplo, debugar um sistema embarcado centrado em software-microprocessador é diferente de debugar um sistema no qual a maior parte do processamento é realizada por periféricos.

Sistemas embarcados freqüentemente residem em máquinas que têm expectativas de funcionamento contínuo por anos, sem erros, ou em alguns casos que se recuperem automaticamente de algum erro que ocorra. Portanto o software é geralmente desenvolvido e testado mais cuidadosamente do que se este fosse destinado a computadores pessoais.

Os softwares embarcados são muitas vezes construídos de forma que um erro possa ser recuperado automaticamente, pois o sistema está inacessível (exemplo: sistemas espaciais), ou precisa estar sempre ligado (sistema de controle de reatores nucleares), ou ainda o mau funcionamento da maquina pode acarretar grandes prejuízos (ex: caixas automáticos).

**5. Metodologia de Projeto**

O projeto de um sistema embarcado consiste basicamente de três fases:

- Análise

- *Design*

- Implementação

Cada etapa consiste em um conjunto de atividades, realizadas de forma não linear. Em geral as etapas de análise, *design* e implementação não são executadas linearmente, uma

abordagem iterativa é a mais conveniente.

Durante a etapa de análise devem ser delineados e documentados os objetivos do sistema.

Na fase de design especifica-se como serão alcançados os objetivos. Na fase de implementação o sistema é construído e testado. Quando se inicia um projeto as informações sobre os objetivos e funcionalidades são, em geral, incompletas. Estas informações, no entanto, permitirão a obtenção de uma primeira arquitetura do sistema. A partir deste ponto pode se partir para a implementação, porém mesmo durante a implementação pode ser necessário voltar para as etapas de análise e design para complementação da funcionalidade ou satisfação de algum requisito.

**Prototipação**

Quando se fala em prototipar um sistema deve se considerar, também, a validação deste, o que em geral representa de 50% a 70% do tempo de projeto. Um fator tem pesado enormemente no desenvolvimento de projetos atuais: a complexidade do projeto do sistema.

Idealmente os projetos deveriam ser desenvolvidos mais rapidamente, lançados no mercado o quanto antes, e devidamente validados. Técnicas de validação de projetos de hardware seguem em geral dois caminhos: a verificação formal e a simulação. Na verificação formal a corretude funcional de um sistema é realizada através de provas matemáticas. A simulação, por sua vez, pode ser usada para verificar a funcionalidade e sua execução requer muitos recursos computacionais.

Infelizmente, o tempo de simulação cresce com o quadrado da complexidade do sistema, o que torna este processo também inviável para grandes sistemas.

Uma possibilidade de se acelerar o processo de validação de projetos é o uso de aceleradores

em *hardware*, capazes de emular grandes sistemas em tempos próximos de suas implementações reais, e que nos permitisse observar aspectos funcionais e temporais dos mesmos. Técnicas de prototipação rápida baseadas em dispositivos reconfiguráveis permitem compensar este gargalo de simulação proporcionando um método de validação rápida durante a fase de desenvolvimento de sistemas, com menor custo computacional e em menos tempo. Ambientes para a prototipação rápida de sistemas digitais tem se tornado uma realidade e estão sendo cada vez mais utilizados graças ao desenvolvimento de dispositivos reconfiguáveis (**FPGA´s**) e das ferramentas de CAD que permitem a síntese de sistemas digitais a partir de sua descrição comportamental.

Os FPGAs surgiram em meados de 1980 como uma nova tecnologia para implementação de circuitos digitais. Estes dispositivos programáveis no campo, eram capazes de implementar uma significante quantidade a mais de *hardware* que os tradicionais PLDs além de lógica multi-níveis. Na sua versão baseada em SRAM, os FPGAs passaram a ser a base para a computação reconfigurável, que tem se tornado uma poderosa metodologia para alcançarmos alta performance na implementação de sistemas digitais. Como foi mencionado, anteriormente, o desenvolvimento de metodologias e ferramentas CAD foram de importância fundamental, para o projeto de sistemas digitais e para a disponibilidade de ambientes de prototipação rápida, em particular.

Um projeto de sistemas digitais, normalmente, envolve uma equipe de pessoas, dividida em grupos onde cada um destes grupos executa uma tarefa diferente no projeto. Sistemas digitais podem ser descritos em três domínios: comportamental, estrutural e físico.

O domínio comportamental, como o próprio nome diz, descreve o comportamento do sistema. A descrição pode ser vista como uma caixa preta, onde se é especificado, apenas, o resultado esperado diante dos valores de entrada. Neste domínio, nenhum detalhe de implementação é dado. O domínio estrutural, ao contrário, define o que está dentro da caixa preta, isto é, a implementação. Neste domínio, descreve-se que elementos funcionais devem ser utilizados, assim como, também, como estes elementos devem estar interconectados. Embora, a partir de uma descrição estrutural, pode-se obter o comportamento do sistema, neste domínio a funcionalidade do sistema não é expressa explicitamente. O domínio físico, descreve as características físicas dos componentes descritos no domínio estrutural.

Uma descrição física expressa a dimensão e a localização de cada componente no chip. Enquanto uma descrição estrutural estabelece a interconexão entre os componentes, uma descrição física define a relação espacial entre os componentes interconectados, estabelecendo o peso, tamanho, dissipação de calor, consumo de potência de cada componente e a posição de cada pino do chip.

Cada domínio pode ter vários níveis de abstração. O nível de abstração de um domínio estabelece qual a complexidade do elemento que se irá trabalhar dentro do domínio. Por exemplo, no domínio estrutural pode-se trabalhar a nível de transistor ou a nível de flip-flops. Neste exemplo, no segundo caso trabalha-se com um nível de abstração maior do que no primeiro caso. Portanto, trabalhar em um determinado nível de abstração corresponde a granularidade com que se deseja trabalhar.

Em 1998, os sistemas embarcados representavam cerca de 10 a 15% do valor de um veículo. A previsão é que chegue a 40% em menos de 5 anos, ou ainda mais para carros elétricos ou de luxo. Carros comuns em 1990 tinham 14 microprocessadores e este ano terão 35. Esta tendência também é comprovada em outras áreas. Por exemplo, helicópteros e aviões militares têm cerca de 60% de seu valor nos sistemas embarcados e apenas 40% em mecânica.

Existe também uma rede de computadores especial para carros, chamada Computer Automotive Network – CAN. Já existem propostas para controlar os dispositivos elétricos, como lâmpadas, por meio de uma rede como esta, de modo a diminuir a quantidade de fios e, consequentemente, o peso dos carros. Imagine que os carros hoje têm freios ABS (Anti-Break System), ignição eletrônica, injeção eletrônica, suspensão ativa, computador de bordo, painel digital (com conta-giros, velocímetro, marcador de combustível, etc), aparelho de som digital, ar-condicionado inteligente (que desliga se o motor for requisitado para uma ultrapassagem, por exemplo), controle de tração, air-bag, alarme contra roubo, entre outros, e todos estes equipamentos são computadorizados.

Equipamentos mais sofisticados já vêm sendo implantados, como sistemas GPS (*Global Positioning System* ou Sistema de Posicionamento Global) que, integrados com mapas digitais, mostram ao motorista a sua localização exata e propoem rotas para o destino desejado. Existem estudos no sentido de fazer com que estes equipamentos possam receber informações sobre o trânsito no percurso desejado de modo a que possam propor rotas alternativas caso haja algum engarrafamento no caminho original. Além disso, estuda-se o uso de carros autônomos capazes de se auto-guiar, os quais aliados a rodovias inteligentes que trocam informações entre os diversos carros, ajuda-os a controlar a distância entre eles e a velocidade, de modo a que se comportem efetivamente como um trem. Cada carro se comporta como um vagão e são controlados automaticamente por computador.

**Wearable Computers ou Ectocomputadores**

Wearable computers são computadores que levamos conosco de uma forma natural. Para isso eles devem poder ser usados enquanto estamos andando. Embora a tendência seja fazer com estes equipamentos tenham um determinado poder de decisão, eles devem sempre permitir o controle do usuário.

No Brasil estes sistemas vêm sendo chamados de computadores vestíveis. Um termo mais apropriado poderia ser ectocomputador, ou seja, computador do corpo. Os ectocomputadores auxiliam seus usuários na interação com o ambiente. O fato de carregarmos estes computadores conosco, permite uma interação muito maior destes com o contexto em que estamos inseridos. Conhecendo melhor este contexto, o ectocomputador pode saber o que fazer sem a ordem direta da pessoa. Para permitir esta interação é comum que estes computadores tenham sensores como câmeras, e dispositivos para comunicação (acesso a telefone e Internet), fones de ouvido e displays miniatura.

Os ectocomputadores já estão sendo testados atualmente. Um exemplo é um computador que traduz a linguagem dos sinais, usada pelos surdos-mudos, para texto, com uma taxa de acerto de quase 100%.Outro exemplo é na área de saúde existem muitas pesquisas sendo feitas mundialmente. No Centro de Informática da UFPE, está sendo desenvolvido, juntamente com o Laboratório de Imunopatologia Keiso Asami (LIKA - UFPE), um sistema com biosensores, os quais serão acoplados a microcomputadores a bateria, num projeto financiado pela FACEPE. Assim o paciente poderá ficar em casa ou no trabalho tendo suas condições clínicas constantemente monitoradas, com uma qualidade de vida muito melhor e, possivelmente, menor custo para os hospitais, em vez de ficar numa cama de hospital coberto de equipamentos. Os dados obtidos pelos biosensores serão transmitidos sem fio pelo ectocomputador para um computador conectado à Internet de modo que o médico responsável possa acompanhar a evolução do quadro clínico ou ser avisado automaticamente em caso de urgência. Evidentemente este monitoramento remoto não se aplica a qualquer caso mas achamos que muitos pacientes poderão se beneficiar de tal tecnologia.

Os maiores desafios tecnológicos no desenvolvimento destes computadores está justamente na percepção do contexto e na tomada de decisões, que envolvem as áreas de processamento de sinais (som, imagem, cheiro, sinais biométricos, etc.) e de inteligência artificial. Também se está estudando como estabelecer a comunicação dos vários ectocomputadores que uma pessoa estiver usando. Assim, os ectocomputadores nos permitirão acesso imediato e constante a informações e outros recursos, como edição de texto, transferência de dados, etc que poderão ser usados para lazer e trabalho a qualquer instante. Porém, deve-se ter bastante responsabilidade ao utilizar os ectocomputadores, já que esses recursos poderão ser usados tanto para a melhoria de nossa qualidade de vida, quanto para restringir nossos diretos. Como toda nova tecnologia, devemos tomar cuidado com sua utilização para que seja usada para o benefício da sociedade.

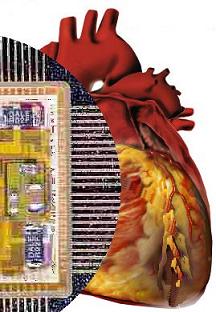
**6. Aplicações**

**Medicina**

Sistemas embarcados médicos usam computadores minituarizados que são designados e programados para fazer apenas uma ou um grande limitado número de funções . O seu tamanho pequeno e o pequeno gasto de energia permitem que eles sejam usados em diversos ambientes.

Scanners de atuais tomografias computadorizadas (TC) utilizam aplicações específicas de circuitos integrados (ASICs) que ocupam algumas polegadas quadradas; Isso é muito grande em comparação com outros sistemas embarcados que cobrem apenas uma mera fração de uma polegada.

Outro sistema embarcado que é muito importante para os scanners de tomografias computadorizadas são os photodiodes (Conversores de luz em corrente elétrica ou voltagem, dependendo da aplicação). Filas de photodiodes (photodiodes arrumados de uma forma linear em um chip integrado ) são responsáveis pela detecção de diferentes comprimentos de onda de luz, uma funçao fundamental na tomografia computadorizada .



Uma mais avançada e recente tecnologia usada para detecção médica é a tomografia de ressonância magnética, que também usa sistemas embarcados. Esse tipo de ressonância requer transmissão de freqüência de rádio e bobinas de recepção, os quais , respectivamente, ativam os sinais e recebem os sinais da ressonância magnética do paciente em observação.

A antena de transmissão de rádio na ressonância magnética são exemplos de aplicações de sistemas embarcados aplicados na área de medicina. Antenas de freqüência de rádio embarcadas ganharam um grande uso com a ampla adoção da técnica de aquisição paralela (usa uma fila de freqüências de rádio para receber dados ) para construções mais rápidas das imagens .

**Sistemas Embarcados no espaço**

Organizações e governos vêm descobrindo diversos usos para sistemas embarcados em seus programas espaciais. Devido ao tamanho diminuto e à baixa demanda de energia, eles se tornaram componentes apropriados em veículos espaciais, onde o espaço físico e a energia disponível são estritamente orçamentados e racionados.



Primeiramente, sistemas embarcados para uso no espaço devem ser duráveis. Eles devem resistir a situações de gravidade zero e a diferentes tipos de radiação. Poucos dispositivos conseguem ser aprovados no rigoroso teste da NASA. Um exemplo de sucesso é o sistema computacional MIP405. Ele é capaz de resistir a aproximadamente 200 milhões de elétron-volts de radiação. Também possui uma vasta gama de temperaturas em que pode operar. Ao mesmo tempo, possui o poder de processamento e as funcionalidades equivalentes às de um laptop antigo.

**Sistemas embarcados e processamento de imagens**

* **Projeto HiRISE**

Uma importante aplicação de sistemas embarcados relacionada a processamento gráfico é o sistema embarcado de alta resolução de imagem de experimentos científicos (HiRISE). Esse sistema está sendo utilizado pela MRO( nave espacial para reconhecimento de marte) . Esse software oferece altíssimas resoluções de imagem tiradas do espaço. Ajudando, dessa forma, a o entendimento dos processos geológicos e climáticos de Marte, tornando possível a identificação de possíveis áreas para aterrissar.

O MRO tem quatro grandes objetivos : determinar se já existiu algum tipo de vida em Marte , caracterizar o clima de Marte , caracterizar a geologia de Marte e identificar locais propícios a estudos científicos. A câmera do HiRISE é de fundamental importância em cada um desses objetivos. HiRISE vai procurar por estruturas geológicas que indicam a presença de água líquida na superfície em algum ponto da historia do planeta. Centenas de locais vão ser examinados com detalhes sem precedentes para revelar minerais relacionados a água e a contribuição da água na formação do terreno. A câmera pode identificar pedras de tamanho tão pequenos como 3 ou 4 pés, ajudando a avaliar a segurança de potenciais áreas de aterrissagem.

HiRISE fornece imagem da superfície de Marte para uma resolução e nível de contraste muito maiores que qualquer imagem anterior. A luz entra pela frente da câmera, é pega por um espelho principal de 50 centímetros de diâmetro e, então, é enviada para uma série de outros espelhos para ser concentrada em detectores.

O método TDI (método utilizado para observação de objetos em alta velocidade, Time Delay and Integration, em inglês ) aumenta a qualidade da imagem, pois a sonda se move há uma velocidade de aproximadamente 3200 metros por segundo. O TDI coleta continuamente e lê os sinais acumulados do CCD (aparelho que transporta sinais analógicos através de sucessivos estágios (capacitores)) .

* **Funcionalidade do sistema embarcado**

O sistema embarcado do HiRISE faz uma serie de diferentes funções que são integradas em um sistema geral de programas. Por exemplo , a câmera é mirada pelo software da sonda (uma aplicação separada), a qual controla thrusthers ( cápsulas que controlam a altitude da sonda no espaço), que posicionam e orientam a sonda para que a câmera esteja apontando para a área que vai ser fotografada. Quando a posição está correta, o software da sonda comanda o software do HiRISE para tirar a foto. O software do HiRISE configura o tempo da linha de pixel do TDI de forma que deixe igual a velocidade do chão e mantenha o alinhamento com o movimento da sonda. O software, então, se volta para a câmera e começa a processar as imagens geradas pelos CCDs , adicionando informações e mandando para o sistema de armazenamento de imagens da câmera, a qual manda as imagens para a estação na Terra. O software do HiRISE também captura informações técnicas da câmera para ajudar a identificar e diagnosticar possíveis problemas. O leitor de sensor fornece informações importantes, como a temperatura dos componentes chaves da câmera. HiRISE avalia essas leituras e envia um aviso se algum parâmetro estiver fora do normal.

A **figura 1** mostra uma foto da superfície de marte tirada pela MRO.

A **figura 2** mostra potenciais áreas de pouso na superfície de marte, as quais foram mapeadas com o auxilio da MRO.

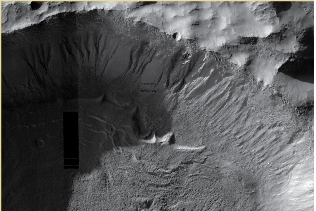


Figura 1. Figura 2 .

Outra aplicação foi o uso de sistemas embarcados no projeto Deep Impact (impacto profundo) da NASA. Esse projeto tinha como objetivo chocar uma sonda em um cometa enquanto outra tirava fotos da colisão e guardava dados importantes sobre a poeira e os gases liberados na explosão(**figura 3**). O código e a tecnologia do software usado na sonda responsável por tirar as fotos foi posteriormente reusados em parte no projeto HiRISE .



**Figura 3**

**Aplicação Militar**

Outro grande exemplo de sistemas embarcados em processamento de imagens é o uso na área militar, onde fatores cruciais contam para a vitória de uma operação.

Um deles é a visibilidade em condições adversas: imagine um soldado que está patrulhando uma área em busca de tanques inimigos. Repentinamente, vê um contingente vindo de uma direção a aproximadamente duas milhas; entretanto, não se pode dizer se são aliados ou inimigos. E então o que ele poderia fazer?

Felizmente, novas tecnologias de aprimoramento de imagens estão disponíveis, que refinam as imagens reais a um nível de detalhes de como se estivessem duas vezes mais perto, clareiam a imagem removendo efeitos de nevoeiros, e aumentam o sinal de contraste em ambientes de pouca luz.

Como funciona?

O software embarcado resolve uma equação matemática, que relaciona a imagem “perfeita” à imagem em questão capturada pelo binóculo do soldado (ou qualquer outro aparelho de vídeo, como uma câmera). Funciona de trás para frente, removendo ruídos e embaçados ao mesmo tempo que ajusta a intensidade de cada pixel, até a obtenção da imagem que mais condiz com dados em tempo real, ou seja, se você sabe como a imagem perde qualidade devido aos efeitos causados pela atmosfera e outros fatores, você pode desfazê-lo.



**GPU**

Uma outra aplicação de sistemas embarcados são no uso de GPU’s embarcadas. **GPU** (**Graphics Processing Unit**, ou **Unidade de Processamento Gráfico**), é o nome dado a um tipo especializado de processamento de gráficos em computadores pessoais, estações de trabalho ou videogames. GPUs modernas manipulam gráficos computadorizados com eficiência e sua estrutura de processamento paralelo os tornam mais capazes neste tipo de trabalho que CPUs normais. Uma GPU normalmente é utilizada em placas de vídeo, mas versões simplificadas são integradas diretamente em placas-mães.

Diferente dos aceleradores gráficos 3D anteriores, uma GPU incorpora as funções de iluminação e transformação de vértices (*transform and lighting* também conhecido como T&L), que antes eram efetuadas no processador central do computador, mais exatamente em seu co-processador matemático (FPU, unidade de ponto flutuante). Seus antecessores efetuavam basicamente a aplicação de texturas em superfícies, e ao incorporar outra etapa do processamento de imagens, libera o processador central para outras funções.

Devido a sua natureza técnica e complexidade, diversos termos em inglês são utilizados para descrever as GPUs, entre os principais se encontram:

* ***Pixel pipeline*** — Parte de um processador gráfico que transfere informações referentes aos *pixels*. Quanto maior a sua quantidade, maior a velocidade da placa em processar pixels para exibição na tela.
  + ***Fillrate*** — Quantidade de *pixels* texturizados que podem ser gerados e exibidos pelo processador gráfico em um segundo.
* ***Render output unit*** (ROP, unidade de saída de renderização) — uma das etapas finais de renderização na qual os *pixels* e *texels* na placa são processados no *pixel* final a ser exibido na tela.
* ***Shader*** — Instruções de *software* usados nos processadores gráficos para renderizar efeitos. Dividem-se em três tipos principais:

***1. Geometry shader*** — Combina uma série de vértices em objetos que serão processados pelos *pixel shaders*.

***2. Pixel shader*** — O *pixel shader* executa rotinas programáveis atuando no nível de *pixel* assim sendo, depois que a imagem foi completamente modelada pelo sistema (CPU e GPU) e teve seus polígonos “ajustados” pelo *vertex shader*, o *pixel shader* entra em ação analisando os materiais dos objetos modelados e as condições de iluminação, e dando um toque final realçando cores onde for necessário, adicionando reflexos onde eles existirem e criando qualquer efeito que o programador quiser com muito mais realismo do que as técnicas mais simples de texturização e iluminação tradicionais.

***3. Vertex shader*** — Manipula vértices e conseqüentemente o formato de objetos. Esses vértices são então enviados para os *geometry shaders*.

* + - ***Unified shader*** — Se refere à capacidade de processar os três diferentes tipos de *shaders* em uma única seção do processador (originalmente cada um dos *shaders* era processado em uma área especializada).
* ***Texture mapping unit*** (TMU, unidade de mapeamento de textura) — Unidade que rotaciona e redimensiona uma imagem bitmap e a aplica em uma superfície ou objeto 3D.
* ***Transform and lighting*** (T&L, transformação e iluminação) — Processos nos quais objetos tridimensionais em um espaço virtual são convertidos em coordenadas bidimensionais para exibição em monitores (transformação) e o cálculo de cores resultantes em objetos iluminados na tela (iluminação). Nome dado geralmente à unidade que acelera estes processos em uma unidade de processamento.



**AMD ATI RADEON E4690 – Nova GPU embarcada da AMD**

**Propósito geral**

São as aplicações mais parecidas com os computadores de mesa, mas em embalagens embarcadas. Nelas costuma haver grande interação entre os usuários e o sistema, geralmente através de terminais de vídeo ou monitores. Como exemplo tem-se os videogames, os conversores de TV a cabo, caixas de bancos, etc.

Sistemas embarcados são usados em controles remotos, diminuindo o preço, o consumo, as dimensões e aumentando a velocidade de resposta e a sua confiabilidade



São também utilizados em controle de sinais e despertadores.



* **Sistemas de controle**

Controles em malha fechada com realimentação em tempo real. Geralmente são as aplicações mais robustas, com placas dedicadas e múltiplos sensores de entrada e saída. Usados nos motores de automóveis, processos químicos, controle de vôo, usinas nucleares, etc.

* **Processamento de sinais**

Envolve um grande volume de informação a ser processada em curto espaço de tempo. Casos de tratamento de áudio, filtros, modems, compressão de vídeo, radares e sonares, etc.

* **Comunicações e redes**

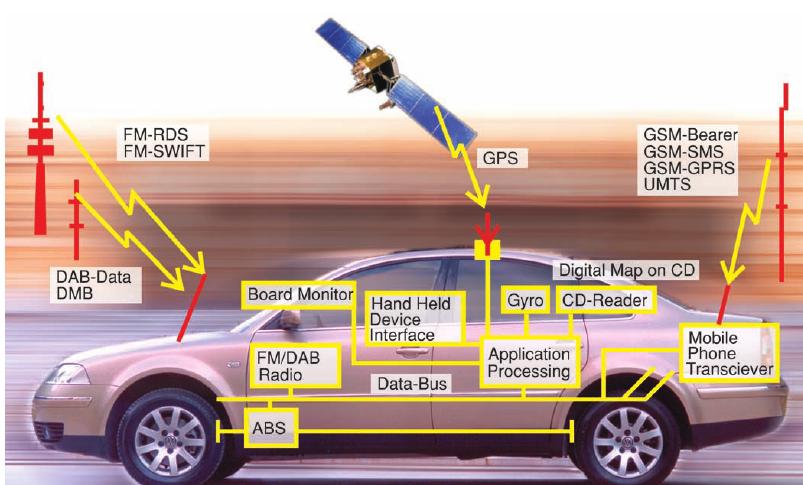
Chaveamento e distribuição de informações. Sistemas de telefonia e telecomunicações e internet.

* **Aplicações embarcadas em chips**

Está se tornando cada vez mais comum o surgimento de chips que consistem num sistema inteiro em uma única pastilha, também conhecidos como SoC (*Systems on Chip)*. É o caso de microcontroladores que já têm embutidos sensores (temperatura, pressão, etc.), transmissores, interfaces gráficas para displays, etc.

* **Aplicações embarcadas no setor automotivo**

Um veículo top de linha é um excelente exemplo de um complexo sistema literalmente “embarcado”. Centenas de sensores fornecem informações sobre todo o funcionamento do veículo. Várias unidades de processamento independentes atuam em regiões diferentes e se comunicam entre si, captando os sinais destes sensores e fazendo com que as ações referentes a cada caso sejam tomadas.  
 Esta comunicação geralmente se dá através de redes. Isto acontece desde a central que memoriza a posição dos bancos, espelhos, volante, etc. para cada usuário do veículo até a central que gerencia o funcionamento do motor.  
 Esta rede de comunicação, além de permitir total interação entre as diferentes áreas internas do veículo, faz com que haja uma grande diminuição da fiação interna, pois todos os elementos se conectam à rede por apenas dois fios, que constituem um barramento de comunicação. A figura abaixo ilustra esta situação.

[](http://www.sabereletronica.com.br/files/image/figura_2_oqsao.jpg)  
**Vários sistemas embarcados em um automóvel**

O uso de sistemas embarcados pode resolver muitos problemas de complexidade e tamanho de produtos elétricos. Um bom exemplo é o do carro acima, no qual vários processos do carro são controlados por dezenas de sistemas embarcados.

**7. Tendências Futuras**

Com o advento dos computadores e da tecnologia da informação em geral (Internet, notebooks, smartphones, etc), os sistemas embarcados estão ganhando força e tornando-se imprescindíveis para o desenvolvimento de novas tecnologias. Assim, eles tem deixado de ser criados apenas como dispositivos isolados, com baixa importância, e têm ganho uma maior abrangência de aplicações. Para tanto, é fundamental incentivar pesquisas acadêmicas e garantir incentivos para investimentos empresariais em prol de assegurar um acúmulo de conhecimentos sobre sistemas embarcados, que possibilite, no futuro próximo, avançar ainda mais nessa nova tecnologia.

Um sonho antigo que está se aproximando cada vez mais com a realidade é a possibilidade de comunicação entre os diversos aparelhos de fabricantes diferentes. Por exemplo, seria prático você poder controlar sua televisão através do celular ou que todos eletrodomésticos de sua casa possam ser manipulados do seu computador, onde quer que você esteja. Já existem inúmeras técnicas desenvolvidas com esta finalidade, como Bluetooth, porém ainda há muito trabalha a ser feito para que haja uma sincronia na atividade em conjunto dos equipamentos. Os sistemas embarcados, em particular, estão sendo desenvolvidos para se adaptar melhor às situação corriqueiras dos usuários, buscando aumentar nossa qualidade de vida na medida que pode ser responsável por facilidades no nosso dia-a-dia. Por isso os sistemas embarcados são tão importantes, sendo decisivos em relação a definição do futuro cenário tecnológico que precensiaremos dentre alguns anos. Enfim, há uma série de fatores que precisam ser melhorados e corrigidos para que os softwares embarcados possam desempenhar o seu papel de forma eficiente, entre os quais se destacam: tamanho, velocidade de processamente, falhas, flexibilidade, custo e energia.

**8. Grupos de pesquisa e projetos brasileiros**

**Grupo de Sistemas Embarcados - GSE**

**Instituição: Pontifícia Universidade Católica do Rio Grande do Sul - PUC RS**

O Grupo de Sistemas Embarcados (GSE) realiza pesquisa e desenvolvimento na área de sistemas embarcados em geral, tanto a nível de software quanto de hardware. As áreas de atuação do grupo incluem aplicações espaciais (sistema de computação de bordo de satélites), mobilidade e pervasive computing, redes wireless ad-hoc (redes de sensores sem fio), tecnologia RFID, computação reconfigurável (VHDL/FPGA), e sistemas embarcados para agricultura de precisão. Entre as atividades de pesquisa em andamento destacam-se os trabalhos em cooperação com: Grupo de Sistemas, Sinais e Computação (SISC) da Faculdade de Engenharia da PUCRS; Grupo de Apoio ao Projeto de Hardware (GAPH), da Faculdade de Informática da PUCRS; Laboratório de Microgravidade do IPCT da PUCRS; Centro de Excelencia em Tecnologia Eletronica Avancada (CEITEC), Space Science Centre da University of Sussex, Inglaterra; Agência Espacial Brasileira (AEB) e Instituto Nacional de Pesquisas Espaciais (INPE). O grupo possui projetos financiados por órgãos de fomento como Finep, Fapergs, CNPq e Agência Espacial Brasileira, incluindo não apenas equipamentos e material permanente, mas também bolsistas de graduação e de pós-graduação. Os resultados das pesquisas do grupo resultam no desenvolvimento de produtos e serviços para aplicação direta pela sociedade. Os resultados mais relevantes das pesquisas são eventualmente aproveitados em produtos comerciais da empresa Innalogics, spin-off do GSE.

**Automação do Projeto de Sistemas Computacionais Embarcados**

Universidade Federal de Santa Catarina - UFSC

O grupo aborda temas das áreas de Microeletrônica, projeto auxiliado por computador (CAD/EDA), bem como software e hardware para sistemas embarcados. O impacto do trabalho deste grupo verifica-se em nove vertentes principais: 1-A modelagem de plataformas de sistemas integrados de hardware e software (SoCs) em vários níveis e estilos de descrição (TLM, funcional/comportamental e RTL). 2-A investigação de novas técnicas de otimização, análise, verificação e síntese a serem incorporadas em ferramentas de Electronic Design Automation (EDA); 3-A investigação de novas técnicas para acelerar o desenvolvimento, a depuração e a otimização de software dependente de hardware, tais como a o redirecionamento automático de montadores, ligadores, depuradores, escalonadores para diferentes processadores-alvo. 4-O desenvolvimento de software dependente de hardware (drivers, componentes de sistema operacional, firmware). 5-A concepção, a verificação e o teste de circuitos integrados digitais (ASICs) e de componentes (IPs) de sistemas integrados de hardware e software (SoCs). 6-O desenvolvimento de técnicas de projeto de circuitos digitais tolerantes a falhas temporárias. 7-O desenvolvimento de arquiteturas para aplicações específicas, em especial a compressão de imagens estáticas e dinâmicas. 8-O desenvolvimento de IPs e drivers voltados para aplicações em dispositivos móveis (em especial Linux-based phones). 9-A prototipação rápida de circuitos e sistemas embarcados através de FPGAs.

**Desenvolvimento e Otimização de Programas para Sistemas Embarcados baseado em Componentes de Software**

Universidade Federal do Amazonas - UFAM

O uso dos chamados componentes de software é um novo paradigma para o desenvolvimento de programas de computador que precisa ser mais bem elaborado de maneira a tornar-se um padrão de desenvolvimento de fato. Nesta abordagem constroem-se componentes de software reutilizáveis, que são amplamente testados, para depois colocá-los à disposição de outros desenvolvedores. De fato, os componentes de software assim utilizados tendem a possuir características de produto, com qualidade assegurada e funcionalidade comprovada. Estatísticas mostram, ainda, o crescente número de sistemas microprocessados que são colocados no mercado mundial a cada ano. De fato, a construção de microprocessadores e sistemas digitais é de fundamental importância para qualquer país que queira uma maior inserção no mercado mundial. Ainda observando o mercado nota-se que apenas aproximadamente dez por cento dos chips construídos são utilizados em computadores de mesa ou laptops. A grande maioria dos chips construídos é utilizada em sistemas menores, sistemas estes com pouco poder de processamento e limitada capacidade de memória. Tais dispositivos são nomeados na literatura de sistemas embarcados ou sistemas embutidos. O tema de interesse deste grupo de pesquisa aborda estas duas áreas do conhecimento tão importantes para a indústria mundial, Componentes de Software e Sistemas Embarcados. Nosso desafio é investigar e propor arquiteturas de componentes de software que tenha aplicações em sistemas embarcados de uso industrial. Para tanto, investigamos tecnologias de componentes de software comerciais que tem uso em sistemas embarcados já relatadas na literatura, as desenvolvidas especificamente para a indústria de sistemas eletrônicos de entretenimento, e as desenvolvidas em institutos universitários de pesquisa. Procuramos validar estas arquiteturas através da da implementação de aplicações reais para sistemas embarcados disponíveis no mercado.

**GPSE - Grupo de Pesquisa em Sistemas Embarcados**

Universidade de Pernambuco - UPE

Os resultados das pesquisas deste grupo abrangem contribuições teóricas e práticas com foco específico em Aplicações Embarcadas. Entre elas constam: Exploração de Arquiteturas e Mecanismos para redução de consumo de energia em aplicações embarcadas.

**Sistemas Embarcados**

Universidade Federal do Rio Grande do Sul - UFRGS

Tendo em vista a crescente dependência da sociedade no uso de recursos computacionais embarcados nos mais diversos sistemas e equipamentos (automóveis, telefones celulares, lavadoras de roupa, instrumentos biomédicos), há uma necessidade crescente de formação de pessoal na linha de pesquisa de projeto e integração completa de sistemas complexos. O grupo busca modelar e realizar automaticamente o projeto de sistemas computacionais embarcados, constituídos de combinações de componentes de hardware e software. Em especial, o trabalho é bastante focado em tecnologias e aplicações que tenham forte possibilidade de apropriação por empresas nacionais, combinado com a produção de trabalhos científicos com impacto internacional. Os tópicos de pesquisa cobrem a arquitetura de processadores dedicados e de sistemas multi-processadores, a geração automática do software embarcado, os sistemas operacionais embarcados de tempo real, as ferramentas de modelagem e validação de sistemas e o teste do software e do hardware dos sistemas embarcados.

**9. Perfil Curricular e Cadeiras Relevantes no Cin**

**Sistemas Embarcados - Cadeiras:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Códi-  go | Área especializante | CH semanal(Teo./Prat) | Crédi-  tos | CH total | Pré-requisitos | Co-requisitos |
| IF725 | Projeto de CPU | 3/2 | 4 | 75 | IF674 | IF827 |
| IF726 | Organização e Tecnologia de Computadores | 3/2 | 4 | 75 | IF674 |  |
| IF727 | Processadores de Aplicação Específica | 3/2 | 4 | 75 | IF827 IF725 IF726 IF709 |  |
| IF728 | Engenharia de Sistemas Embutidos | 3/2 | 4 | 75 | IF674 |  |
| IF709 | Implementação de Sistemas Operacionais | 3/2 | 4 | 75 | IF677 IF674 |  |
| IF729 | Prototipação de Circuitos Integrados | 3/2 | 4 | 75 | IF674 |  |
| IF730 | Sistemas de Tempo Real | 5/0 | 5 | 75 | IF677 |  |
| IF814 | Met. de Hardware/Software Co-design | 3/2 | 4 | 75 | IF674 |  |
| IF732 | Projeto de Sistemas Embutidos | 3/2 | 4 | 75 | IF729  IF728 |  |
| IF733 | Transdutores | 3/2 | 4 | 75 | ES238 |  |
| IF828 | Validação de Sistemas Embutidos | 3/2 | 4 | 75 | IF674 |  |
| IF829 | Tolerância a Falhas | 3/2 | 4 | 75 | IF674  IF677 |  |
| IF830 | Sistemas Assíncronos | 3/2 | 4 | 75 |  |  |
| IF831 | Arquiteturas Avançadas de Computadores | 5/0 | 5 | 75 | IF674 |  |
| IF832 | Medidas de radiofreqüência | 3/2 | 4 | 75 | FI108 |  |
| IF833 | Instrumentação Eletrônica | 3/2 | 4 | 75 | ES238 |  |
| IF735 | Tópicos Avançados em Arquitetura de Computadores | 5/0 | 5 | 75 | IF674 |  |
| IF737 | Tópicos Avançados em Sistemas Embutidos | 5/0 | 5 | 75 | IF674 |  |
| IF736 | Seminário em Arquitetura e Sistemas Embutidos | 3/0 | 3 | 45 | IF674 |  |

**10. Conclusão**

Como se pode notar, apesar dos Sistemas Embarcados não serem uma tendência muito moderna no campo da tecnologia (eles existem há mais de meio século), eles representam uma aplicação de crescente importância no mundo atual. Assim como a Internet, que foi desenvolvida para um propósito específico, os Sistemas Embarcados foram criados sem pensar na imensa gama de aplicações que eles podem ter. Hoje em dia esses sistemas se tornaram regra ao invés de exceção, e representam as verdadeiras aplicações da computação no mundo atual, estando presentes desde o controle remoto da televisão até o sistema de controle de um ônibus espacial da NASA.

Mesmo com o que foi descoberto sobre esses sistemas, ainda há muito que se fazer. No futuro, o grande propósito das pesquisas nessa área é desenvolver tecnologia para interligar diferentes sistemas e aumentar a capacidade funcional desse conjunto, proporcionando assim maior facilidade e comodidade no desempenho das tarefas para qual foram projetados. Assim sendo, no futuro, os produtos terão seus preços determinados pelas tecnologias embarcadas presentes neles. E aqueles que souberem dominar essa tecnologia serão aqueles que determinarão os preços dos produtos, e lucrarão com isso.

**11. Bibliografia**

* Embedded System Design: A Unified Hardware/Software Introduction–Frank Vahid and Tony Givargis, John Wiley and Sons, 2002
* http://stefanor.uctleg.net/course-notes-archive/eee3074w/EEE3074W-Lecture02.pdf
* <http://www.ad-mkt-review.com/public_html/air/ai200205.html>
* <http://en.wikipedia.org/wiki/Embedded_systems>
* <http://pt.wikipedia.org/wiki/Sistema_embarcado>
* <http://en.wikipedia.org/wiki/Apollo_Guidance_Computer>
* <http://www.sabereletronica.com.br/secoes/leitura/274>
* [http://www.mil-embedded.com/](http://www.mil-embedded.com/articles/id/?2840)
* <http://www.scienceprog.com/>
  + http://www.guiadohardware.net/artigos/entendendo-sistemasembarcados
* Programming embedded systems in C and C++, editora: O'reilly , autor: Michael Barr
* <http://www.neoradix.com.br/arq/embedded/01_sysembedded.pdf>