

Systems on Chips Multiprocessados: MPSoCs

Julian Pontes
Sérgio Johann Filho
Valderi Leithardt

Sumário

- I Introdução a SoCs
- I IPs reutilizáveis
- I Software embarcado
- I Projeto baseado em plataformas
- I Estruturas de comunicação intra-chip
- I MPSoCs
- I Estudo de caso em MPSoCs

Introdução a SoCs

- I A indústria de semicondutores tem aumentado em muito a densidade (transistores) de circuitos integrados em larga escala (VLSI)
- I Para conseguir coordenar a utilização dos níveis de integração disponíveis, novas metodologias têm sido aplicadas por projetistas
 - Metodologia emergente: SoCs – consistem de blocos (*IP cores*) pré verificados e pré validados, projetados internamente em uma organização ou obtido de terceiros, integrados em um único *chip*
- I Exemplos de IPs: processadores embarcados, módulos de memória, *interfaces*, blocos analógicos, DSPs
- I Componentes de *software* para os módulos também são reutilizáveis (RTOSs, bibliotecas, *drivers*)

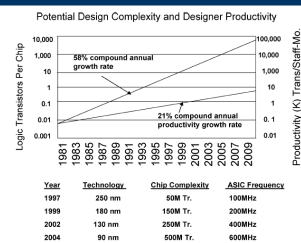
Introdução a SoCs

- I Implementações em SoCs – grandes ganhos de produtividade
- I Papel do projetista de SoCs
 - Integrar funções (IPs) para formar sistemas mais complexos
 - Projeto relativamente rápido
- I Processo de integração
 - Conexão de blocos ao meio de interconexão
 - Grandes ganhos são possíveis quando o sistema é modelado como uma plataforma, onde o projeto é realizado com o intuito de gerar derivações rapidamente
 - Reuso
 - Redução de consumo de energia
 - Miniaturização
 - Diminuição dos custos de produção

Introdução a SoCs

- I Integração de mais funcionalidades em um único *chip* tenta cumprir com a lei de Moore, que prevê que o número de transistores em um *chip* irá dobrar a cada 18-24 meses
- I Desafio: aumento da produtividade para que a meta da lei de Moore seja atingida
- I Noção de SoC: ganhos de produtividade através do reuso de projeto e integração de componentes

Introdução a SoCs



IPs reutilizáveis

- I Pré-requisito para a criação de SoCs – conjunto de blocos reutilizáveis e que suportem a integração em um sistema de forma transparente (padronização de *interfaces*)
- I Biblioteca de blocos reutilizáveis em diversas configurações com relação à frequência, área e consumo de energia é a chave para o sucesso no desenvolvimento de sistemas
 - Projetista pode integrar blocos em diversas configurações e avaliar o custo / desempenho para a aplicação alvo
- I Problemas
 - Processo de criação de blocos reutilizáveis difere da implementação tradicional de ASICs
 - Tipicamente, 5 vezes mais trabalho é realizado na concepção de blocos reutilizáveis se comparados a blocos específicos

IPs reutilizáveis

- I A forma de um IP reutilizável pode variar dependendo da forma que o desenvolvedor do IP escolhe prover ao projetista
 - Três categorias principais: *soft, firm, hard*
- I Blocos *soft IP* – especificados em RTL ou outra descrição de alto nível
 - HDLs são independentes de tecnologia / processo e podem ser sintetizados para um nível de portas lógicas
 - Vantagens: flexibilidade, portabilidade, reusabilidade
 - Desvantagens: não existem garantias de temporização nem de consumo de energia, pois diferentes processos de fabricação e aplicação possuem desempenhos diferentes
 - Muitas vezes o código HDL é encriptado para evitar modificações

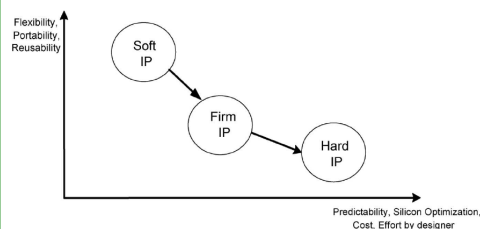
IPs reutilizáveis

- Blocos *soft IP* podem ser modificados para uma determinada aplicação. Existem ferramentas que otimizam determinada descrição, e o *hardware* gerado no final do processo pode ser mais eficiente que uma implementação *hard*
- I Blocos *hard IP* – possuem *layout* fixo e são altamente otimizados para uma determinada aplicação em um processo específico
 - Vantagens: possuem desempenho bem definido, são pré-qualificados e já foram previamente testados em silício
 - Desvantagens: alto custo e grande esforço no seu desenvolvimento além da pouca portabilidade entre diferentes processos / tecnologias

IPs reutilizáveis

- I Blocos *firm IP* – são descrições de circuitos parametrizáveis, onde o projetista pode otimizar o desempenho do bloco para uma aplicação específica
 - Os parâmetros flexíveis permitem ao projetista prever de melhor forma o desempenho de determinado bloco
 - Esse tipo de IP oferece uma relação de compromisso entre *soft* e *hard*, sendo mais flexível e portátil que um *hard IP* e mais previsível que um *soft IP*

IPs reutilizáveis



Software embarcado

- I *Software* é a maneira mais natural de programabilidade para um SoC
- I O *software* embarcado permite que um único SoC possua diferentes "personalidades", servindo diferentes consumidores e segmentos de mercado
 - De uma perspectiva de projeto, quanto mais *software* em um sistema, maior a flexibilidade do mesmo
- I Operações de controle complexas são mais facilmente implementadas em *software*, enquanto operações em dados são mais facilmente implementadas em *hardware*

Software embarcado

- I Reuso é provido pelo uso de bibliotecas e estruturas de dados
- I Pode ser atualizado, ou seja, está em um desenvolvimento contínuo
- I A programabilidade de SoCs caminha nessa direção, dessa forma:
 - Aumento da necessidade de projetistas de *software / firmware* em uma indústria tipicamente dominada por engenheiros de *hardware*

Projeto baseado em plataformas

- I Ganhos de produtividade obtidos estritamente através de reuso de IPs tem seus limites
 - Processo de integração de um *chip* pode levar muito tempo
 - Custo de projeto pode ser elevado, mesmo com blocos pré desenvolvidos
- I Para aumentar a produtividade, um maior nível de abstração torna-se necessário
 - Projeto baseado em plataformas
 - Uma plataforma pode ser utilizada para diversos projetos
 - Plataforma: possui um nível de abstração alto, e decisões de alto e baixo nível já estão embutidas e tomadas, diminuindo o tempo de projeto
 - Ferramentas e fluxos para otimizar o projeto

Projeto baseado em plataformas

- I Uma plataforma consiste de *hardware* IPs, *software* IPs, estruturas de interconexão padronizadas (AMBA, CoreConnect, etc.), ferramentas CAD para desenvolvimento integrado *hardware / software*, implementação, verificação e validação, geração de derivações da plataforma e *hardware* dedicado à prototipação
 - Exemplo: Ferramentas EDK e ISE (Xilinx)
 - Outro exemplo: plataforma bluetooth (University of British Columbia)

Projeto baseado em plataformas

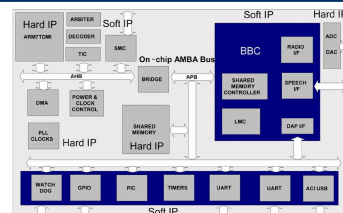


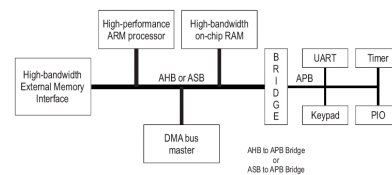
Fig. 10. Bluetooth hardware platform.

Estruturas de comunicação intra-chip

- I Atualmente, sistemas MPSoC consistem em diversos núcleos operando em frequências diferentes, o que dificulta a sua integração
 - Diferenças de frequência entre um DSP e uma interface Ethernet
 - Diferentes núcleos são projetados para trabalharem em frequências diferentes, e são fabricados por várias empresas
- I Lógica tradicional – um único relógio global
- I Linguagens de modelagem a nível de sistema (SystemC, System Verilog) – conceito de processos comunicantes
 - Cada núcleo é visto como uma ilha, e se comunica através de transações
 - GALS

Estruturas de comunicação intra-chip

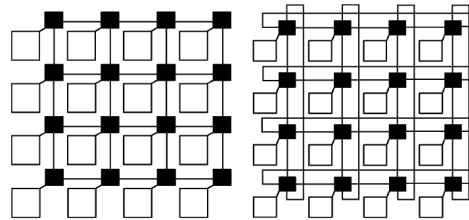
- I Problema: como garantir a latência de comunicação em um sistema GALS, para um correto funcionamento do sistema?
 - Tema contínuo de pesquisas
- I Barramentos hierárquicos



Estruturas de comunicação intra-chip

- I Redes intra chip (NoCs)
 - Meio mais estruturado de intercomunicação entre núcleos
 - Atualmente, entre 10 e 100 (número grande de IPs)
 - Muito usada em MPSoCs com grande número de processadores
 - Sistemas que utilizam NoCs como meio de comunicação funcionam como verdadeiros multiprocessadores
 - Novas metodologias devem ser adotadas para implementação de sistemas desse tipo, pois existem restrições no consumo de energia e comunicação entre IPs
 - Surgimento de novos problemas
 - Pesquisas recentes indicam que NoCs dominarão o mercado de MPSoCs, pois esses serão "mares de processadores"

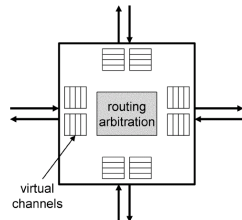
Estruturas de comunicação intra-chip



(a) Topologia malha

(b) Topologia Toro

Estruturas de comunicação intra-chip



Estruturas de comunicação intra-chip

- I NoC: arquitetura de comunicação que naturalmente implementa *pipelines* profundos
- I Múltiplas comunicações em paralelo
- I Comunicação através de chaveamento de circuito ou comunicação por pacotes
 - Diversas técnicas de transmissão de pacotes: *store-and-forward*, *virtual cut-through*, *wormhole*

MPSoCs

- I Muitas vezes, a única forma de se implementar um sistema com aplicações complexas em um tempo razoável é utilizar componentes programáveis
- I Sistemas embarcados monoprocesados: típicas aplicações implementadas com microcontroladores – baixo desempenho
- I Novas aplicações: requerem alto poder de processamento para alcançar o desempenho necessário
- I MPSoCs estão sendo cada vez mais utilizados para a implementação de aplicações de alto poder de processamento

MPSoCs

- I MPSoC: mais do que apenas múltiplos processadores interconectados em um único *chip*
 - Requerimentos de aplicação e de implementação obrigam o projeto a ter uma arquitetura heterogênea
 - Altas taxas de computação, e qualidade de serviço
 - Baixo consumo de energia
 - Baixo custo
- I MPSoCs balançam esses requerimentos, adaptando a arquitetura às necessidades de uma aplicação
 - Usar o poder de processamento onde ele seja necessário
 - Eliminar elementos desnecessários, para baixar o consumo de energia e o custo

MPSoCs

- I MPSoCs x Chip Multiprocessors
 - *Chip Multiprocessors* utilizam o aumento da densidade de transistores para terem mais processadores em um *chip*
 - MPSoCs são arquiteturas que balanceiam as limitações da tecnologia VLSI de acordo com as necessidades da aplicação
- I Elementos de processamento especializados
 - Instruções especializadas são usadas para acelerar determinado processamento, e prover ainda assim flexibilidade
 - Dificuldade na geração de CPUs especializadas, além das ferramentas (compiladores, *debuggers* ...)

MPSoCs

- I Processadores configuráveis – projetados para serem extensíveis em diversos aspectos
 - Conjunto de instruções
 - Tamanho de palavra
 - Tamanho das *caches*
- I Conjunto de refinamentos da arquitetura são usados para a geração do *hardware*, compilador entre outros – ferramentas
- I Alternativa à CPUs configuráveis: aceleradores dedicados
 - Funções usadas regularmente e com grandes quantidades de dados
 - Exemplo: DCT (transformada do cosseno) usada em algoritmos como MPEG2, MPEG4, JPEG
 - Muito útil quando não existe necessidade de flexibilidade

MPSoCs

- I Sistemas de memória
 - Sistema heterogêneo é bastante usado
 - Blocos de memória especializados (pequenos)
 - Blocos de memória acessíveis à toda plataforma (grandes)
- I Heterogeneidade
 - Dificulta a programação
 - Necessária para a implementação de MPSoCs
- I Porque utilizar memórias especializadas?
 - Desempenho *real-time*
 - Apenas um ou poucos processadores acessam a memória especializada, tornando sua latência de acesso mais previsível
 - Previsibilidade

MPSoCs

- Arquiteturas de memória especializada consomem menos energia
- Barramentos menores consomem menos energia
- I Estruturas de interconexão:
 - Barramento – muitos trabalhos direcionados à arquitetura, arbitragem. Problema: escalabilidade
 - NoC – distribuição de múltiplas rotas de comunicação. É uma tendência, pois é mais flexível e escalável – maior variedade de produtos MPSoC derivados

MPSoCs

- I Desafios e oportunidades
 - A lei de Moore apresenta desafios contínuos
 - MPSoCs – combinam as dificuldades na construção de sistemas complexos tanto de *hardware* quanto de *software*
 - Metodologias para o desenvolvimento tornam-se necessárias
- I Metodologias que funcionam
 - Diminuição do tempo que um produto leva para estar pronto
 - Torna mais fácil uma previsão de quanto tempo será o desenvolvimento de um produto
 - Melhorias de desempenho e diminuição do consumo de energia podem ser aplicados em diferentes projetos
 - Metodologias serão cada vez mais necessárias, tendo-se em vista projetos mais otimizados – novas ferramentas e abordagens

MPSoCs

- I *Hardware / software codesign*
 - Elemento chave para o desenvolvimento – continuidade entre *software* e *hardware*
 - Modelos de programação em alto nível que servem para esconder interfaces sofisticadas de HW / SW
- I Necessidade de melhorias no projeto
 - Projeto concorrente de HW / SW embarcado, levando a um menor *time-to-market*
 - Projeto modularizado de componentes HW / SW, aumentando a clareza nas escolhas do projeto na construção de sistemas complexos
 - Validação global mais fácil para todo o sistema

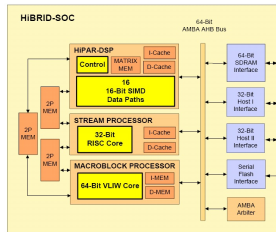
MPSoCs

- I HW / SW *codesign*
 - Melhoria na qualidade do *software* embarcado
 - Interfaces abstratas de HW / SW possibilitam uma validação mais rápida
 - Interfaces abstratas permitem o uso de padrões
 - Aumento da interoperabilidade
- I Uso de ferramentas

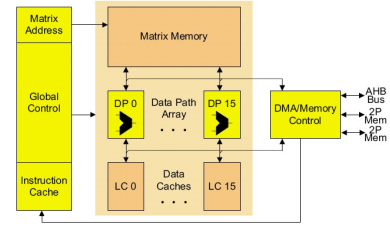
Estudo de caso

- I Hibrid-SoC – uma arquitetura SoC multiprocessada para aplicações de processamento de sinais multimídia
- I Alto poder de processamento multimídia e alta flexibilidade – programabilidade por *software*
- I Aplicação principal – MPEG4 de alta resolução
- I Plataforma extensível

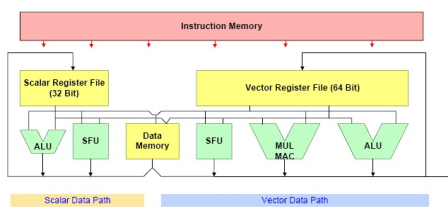
Estudo de caso



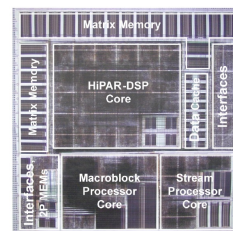
Estudo de caso



Estudo de caso



Estudo de caso



Estudo de caso

Table 2. MPEG-4 ASP decoder performance on MP and SP, 720×576@25Hz, 1.5–3 Mbit/s:

| <i>Task</i> | <i>MHz</i> |
|---------------------|------------|
| IDCT | 10.0 |
| Motion Compensation | 100.4 |
| Reconstruction | 5.3 |
| Other | 4.0 |
| Overall | 119.7 |