

# ESTUDO DO MICROCONTROLADOR IHM-PLASMA

Lars Bleckwedel Morosini<sup>1</sup>, Prof. Salvador Pinillos Gimenez<sup>2</sup>

<sup>1,2</sup> Engenharia Elétrica, Centro Universitário da Fundação Educacional Inaciana  
[larsbleckwedel@outlook.com](mailto:larsbleckwedel@outlook.com), [sgimenez@fei.edu.br](mailto:sgimenez@fei.edu.br)

**Resumo:** Este artigo descreve o estudo do microprocessador Plasma, que foi projetado baseado nos conceitos de processamento paralelo, para conter parte do Sistema Operacional de Tempo Real (*Real Time Operating System, RTOS*), que neste caso foram implementados os processos de agendamento (*scheduler*) e gerenciamento de contexto (*Context Manager*). Esse inovador microprocessador foi intitulado de *Interlocked Hardware Microkernel-Plasma* (IHM-Plasma). O IHM-Plasma tem a capacidade de melhorar o desempenho em processamento em até 151% em relação àquele que o originou.

## 1. Introdução

Para os processadores que temos na atualidade, foram-se necessários muitos anos de pesquisa e estudo para aperfeiçoar a tecnologia que hoje consideramos ter aberto muitas portas que a humanidade, até poucas décadas, considerava impossíveis de se realizar. Não é à toa que com tamanho potencial, acabou-se cobrando mais adaptabilidade para sistemas ainda rígidos e com pouca lapidação. Um exemplo claro é visto em sistemas controlados que necessitam de um tempo de processamento maior, mas que por poucas vezes acabam não entregando a resposta desejada pelas suas próprias limitações de *hardware* [1].

Neste cenário, este trabalho de Iniciação Científica vem fazer o estudo dos trabalhos realizados pelos pesquisadores Dr. Salvador Pinillos Gimenez et. al. [1], Igor Krause [2] e Prof. Leandro Poloni Dantas [3], que propuseram uma arquitetura inovadora baseada no processador convencional, visando aprimorar ainda mais esta tecnologia. A técnica utilizada para tal feito vem da utilização de um *hardware* auxiliar, que proporciona uma troca entre o processamento antes feito pela própria unidade central de processamento (*Central Processing Unit, CPU*) para este *hardware*. Nisto o sistema ganha em processamento voltado em tarefas para minimizar o tempo gasto pela CPU executando atividades paralelas que não envolvessem a tarefa de fato [1]. O nome que os autores [1] deram para este novo processador foi “*Interlocked Hardware Microkernel-Plasma*” (IHM-Plasma).

## 2. Descrição de hardware do IHM-Plasma

Em sistemas baseados em tarefas, que é o caso para sistemas operacionais de tempo real (*real-time operating systems, RTOS*) objeto de estudo desta IC, o tempo que o processador gasta atualizando as tarefas, trocando os contextos e gerenciando processos envolvendo atualizações do sistema, interfere diretamente no tempo cedido para completar a tarefa exigida e torna-se mais crítico este tempo quanto menor

é o período para executar tal tarefa [1]. Para este caso, e outros envolvendo prazos curtos, eram oferecidos tradicionalmente duas opções para sanar o problema: a primeira é aumentar o *clock* (frequência) da CPU, ou aumentar a quantidade de núcleos (unidades de processamento) da CPU [1].

Com esta nova implementação, analisemos para o dado sistema fornecido em [1] na Figura 1, as principais modificações feitas com a implementação do sistema do *hardware* IHM-Plasma:

### 1) 2 Bancos de Registradores:

No lugar de um banco de registradores interno (*Internal Register Bank, IRB*) foi acoplado um outro idêntico [Block (a) na Fig.1]. A proposta é que enquanto um IRB estiver sendo utilizado pela CPU para executar a tarefa atual, o outro está sendo acessado pelo *hardware* auxiliar para atualizar o contexto da tarefa atual e carregar o contexto da próxima. Desta forma, quando o tempo de execução (*time slice*) terminar os IRB's trocam de função [1].

### 2) Timing & Control (T&C)

O bloco denominado de *Timing & Control* (T&C) [Bloco (e) na Fig.1] é responsável por fazer todas as comunicações dentro do IHM. Além disso, também controla o fluxo de operações do *hardware*, o *time slice*, o estado atual de cada tarefa etc. [1].

### 3) Hardware Scheduler (HS)

O *Hardware Scheduler* [Bloco (f) na Fig.1], assim como o próprio nome sugere, define a próxima tarefa que a CPU irá executar [1].

### 4) Context Manager (CM)

O *Context Manager* [Bloco (g) na Fig.1] recebe dois tipos de sinais de ativação, o *backup* e o *restore*. O sinal “*backup*” é responsável por copiar o contexto da tarefa atual, e o “*restore*” carrega para o IRB o contexto da próxima tarefa a ser executada [1].

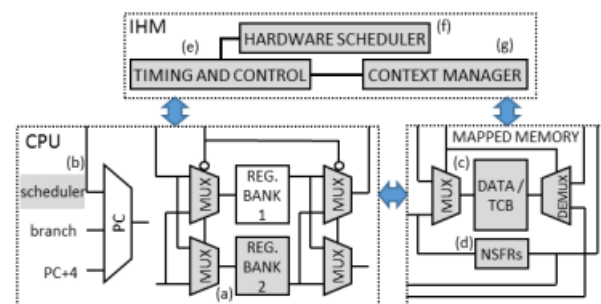


Figura 1 – Diagrama de Blocos simplificado do processador customizado proposto [1].

## 3. Análise dos Resultados Previamente Obtidos

Dentro das pesquisas realizadas pelos estudos mencionados vale ressaltar alguns resultados obtidos.

No primeiro teste, feito no trabalho [1], um programa escrito em linguagem C foi executado nos dois processadores, onde foram utilizadas duas tarefas que estavam com a mesma prioridade de execução. Deste teste criou-se a Tabela 1 [1].

A primeira parte da tabela apresenta o número de cc necessários para realizar as tarefas de mudança de contexto, escalonamento e processo. A segunda parte apresenta a quantidade de recursos de *hardware* e *clock* máximo da família *Intel FPGA Cyclone V* que foram utilizados para implementar o Plasma original e o IHM-Plasma [1].

Tabela 1 - Quantidade de ciclos de *clock* e *hardware* usados pelo IHM-Plasma e o Plasma original [1].

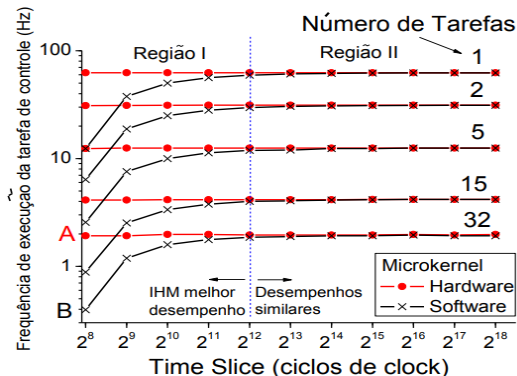
Item	IHM-Plasma	Plasma	IHM-Plasma Gain
Context switching (cc)	2	135	-98,5%
Scheduling (cc)	0	20	Infinite
CPU processing a task (cc)	254	101	+151%
Total logic used in adaptive logic modules	2,250	1,263	+77%
Total number of registers	1667	504	+231%
Total block memory bits	67,584	67,584	0%
Maximum clock rate (MHz)	33,26	43,33	-23,6%

De acordo com a Tabela 1, podemos notar o ganho significativo (+151%) de cc utilizados pela CPU para execução de tarefas, além de um ganho infinitesimal com relação ao cc de contextualização e uma diminuição tremenda (-98,5%) no cc para mudar o contexto para a próxima tarefa. Os resultados obtidos foram graças ao aumento (+77%) no total de módulos de lógica adaptativa adicionais e um aumento (+231%) no total do número de registradores, bem expressivos se levado em consideração o modelo original [1].

Outra pesquisa com grandes diferenças entre a abordagem utilizada pelos pesquisadores e a abordagem tradicional pode ser analisada na Figura 2 [3].

Neste experimento, foi-se diminuindo gradualmente o *time slice*. No eixo cartesiano “x” do gráfico temos o *time slice*, e no eixo cartesiano “y” temos a frequência em *Hertz* (Hz). A quantidade de tarefas ativas está mencionada na ponta das curvas [3].

Figura 50 – Frequência de execução de uma tarefa de controle (*taskControl*) em função do *time slice*, considerando-se diferentes números de tarefas ativas, quando o escalonamento é realizado por algoritmo em *hardware* (IHM) e em *software* (abordagem tradicional) [3].



Como demonstra a Figura 2 [3], com a crescente quantidade de tarefas ativas para sempre trabalhadas pela CPU, menor fica a frequência de processamento das mesmas, isto em ambas as abordagens [3]. O que se observa de peculiar é como os sistemas se comportam a medida que a quantidade de cc disponíveis diminui, no caso do sistema desenvolvido a tendência se comporta linearmente. Em contra partida a técnica convencional tem um declínio curvilíneo a partir de um certo ponto, nele o gráfico foi dividido no desempenho similar e onde há um ganho considerável pela técnica do IHM-Plasma. Esta diferença se deve por conta do tempo de troca de contexto. Na abordagem estudada este tempo se resume em 2 cc, como analisado na Tabela 1, portanto o sistema pode focar inteiramente na execução da tarefa sem precisar interromper o processo e decair no desempenho conforme se diminui o *time slice* [3].

A partir desse estudo realizado, os próximos passos envolvem fazer simulações testando a tecnologia aplicada e futuramente implementar programas em C utilizando destes conceitos para serem rodados no processador IHM-Plasma.

#### 4. Conclusões

Para a proposta inicial do projeto, pode-se afirmar que de fato houve uma resposta significativamente superior as abordagens utilizadas até então, levando-se em consideração a quantidade de performance em execução de tarefas gerado pela tecnologia inovadora. E para esta proposta de Iniciação Científica, portanto, cabe de continuar os avanços neste campo de estudo a fim de desenvolver ao máximo os recursos implementados, para que futuramente possa-se continuar a quebrar barreiras que antes eram consideradas impossíveis.

#### 5. Referências

- [1] Dantas, L. P. et al. (2019). A Novel Processor Architecture With a Hardware Microkernel to Improve the Performance of Task-Based Systems. *IEEE Embedded Systems Letters*, 46-49.
- [2] Krause, I. (2022). *Estudo da Implementação de um Escalonador Baseado no Algoritmo Earliest-Deadline-First por Hardware para o IHM-Plasma Usando Conceitos de Processamento Paralelo*. São Bernardo do Campo.
- [3] Dantas, L. P. (2018). *Desenvolvimento de um Inovador Microkernel por Hardware para Microcontroladores Focando nas Aplicações de Sistemas Baseados em Tarefas*. São Bernardo do Campo.