

μ PD - Microprocessador de 32 bits com funções de Processamento Digital de Sinais para uso didático

Amanda Mendes Ferreira Gomes¹, Matheus Trierveiler¹, Marcelo Daniel Berejuck²

¹ Departamento de Engenharia Elétrica - Florianópolis
Universidade Federal de Santa Catarina (UFSC)

² Departamento de Computação - Araranguá
Universidade Federal de Santa Catarina (UFSC)

Abstract. *This paper introduces the organization and architecture of a 32-bit microprocessor, called μ PD. It was designed to be employed as support in structural disciplines of Electrical Engineering and Computer Science, as "Microcontrollers and Microprocessors," and "Computer Architecture." It also was designed to be adopted in teaching disciplines related to digital signal processing. It was implemented in synthesis language (VHDL) using some machine instructions of the MIPS 32 processor. This document will introduce the functional blocks that comprise the organization of the microprocessor, its architecture, and the silicon consumption due to its implementation of Field Programmable Gate Array technology.*

Resumo. *Este artigo apresenta a organização e arquitetura e um processador de 32 bits, chamado μ PD. Ele foi projetado para ser usado como ferramenta de suporte em disciplinas estruturantes dos cursos de Engenharia Elétrica e Engenharia de Computação, como "Microcontroladores e Microprocessadores" e "Arquitetura de Computadores". Ele também foi projetado para ser utilizado em disciplinas relacionadas a processamento digital de sinais. Sua implementação foi feita em linguagem de síntese de hardware VHDL utilizando algumas instruções do processador MIPS 32. Este documento apresenta os blocos funcionais que compõem a organização do processador, sua arquitetura, e o consumo de silício devido a implementação em um dispositivo do tipo FPGA (Field Programmable Gate Array).*

1. Introdução

O conhecimento do funcionamento de um sistema embarcado baseado no uso de microprocessadores ou microcontroladores é essencial para que um estudante de Engenharia Elétrica ou de Computação possa compreender muitos conceitos da área de programação. Percebe-se que os conceitos abordados em disciplinas relacionadas a Algoritmos e Programação são melhores compreendidos pelos alunos quando eles têm a oportunidade de associar os conceitos das linguagens de programação com a organização interna do processador.

Trabalhos relacionados a processadores simplificados e implementados em linguagem de síntese vem sendo publicados na comunidade científica, tais como [Vieira et al. 2016] [Nooshabadi and Garside 2005]. O foco de tais trabalhos foi o uso daqueles processadores em disciplinas estruturantes de Engenharia Elétrica e

Computação, como “Microcontroladores e Microprocessadores” ou “Arquitetura de Computadores”. Porém, não foram encontradas até o momento publicações de processadores básicos, baseadas em linguagem de síntese, que também possam ser adotadas em disciplinas de Processamento Digital de Sinais.

Um processador simplificado que apresente funções específicas de processamento digital de sinais permite aos alunos de graduação a realização de análises mais apuradas sobre o desempenho de algoritmos específicos desta área do conhecimento, como filtros digitais sendo realizados em software ou hardware. Nesse contexto, este artigo apresenta os resultados do projeto de desenvolvimento de um processador para uso didático que visou o desenvolvimento de um processador com um conjunto de instruções mínimo para auxiliar o aprendizado de conceitos de Arquitetura e Organização de Computadores por alunos em fases iniciais do curso de graduação em Computação e Engenharia Elétrica, além de oferecer instruções específicas relacionadas ao processamento digital de sinais e servirem como ferramentas para a execução de experimentos em disciplinas relacionadas à este tema.

Este documento está organizado da seguinte forma: A Seção 2 apresenta uma visão sucinta de trabalhos realizados no ensino de processadores para Engenharia e Computação. A Seção 3 apresenta a arquitetura do processador, com os opcodes implementados além dos novos *opcodes* definidos especificamente para aplicações de processamento digital de sinais. A Seção 4 descreve os principais componentes desenvolvidos para o processador μ PD. A Seção 5 apresenta os resultados experimentais obtidos com a ferramenta de síntese do fabricante *Xilinx*. Por fim, a Seção 6 apresenta as nossas conclusões e expectativas de trabalhos futuros.

2. Trabalhos relacionados

Com a constante evolução no desenvolvimento de microcontroladores e microprocessadores para uso em sistemas embarcados, a escolha de arquiteturas de processadores para serem usados como referência no ensino de Arquitetura e Organização de Computadores para alunos de graduação é tema frequente de estudos por educadores da área. Autores como [Clements 1999] discutem os aspectos que devem ser levados em consideração na escolha de modelos de processadores a serem aplicados no ensino de alunos de graduação.

Uma das primeiras iniciativas de sucesso com uso de microprocessador didático foi o processador denominado de MU0, desenvolvido pela *University of Manchester* [Furber 2000]. Ele é um processador simples, baseado no modelo Von Neumann, composto dos seguintes componentes:

- (i) um contador de programa (PC) que registra o endereço da instrução em execução;
- (ii) um registrador chamado de “acumulador” (ACC) que armazena o valor de um dado durante a execução de uma instrução; e
- (iii) uma unidade lógica e aritmética que utiliza os registradores PC e ACC afim de realizar as operações indicadas nas instruções de máquina.

O uso de apenas um único registrador para realizar as operações básicas de um processador didático também pode ser observado nos simuladores Neander-X, Ahmes e Ramses apresentados por [Weber 2012].

O Neander-X é um processador de 8 bits cujas operações (soma, lógica AND e lógica OR) são realizadas sobre um registrador denominado de “acumulador” (ACC). O processador Ahmes é baseado no processador Neander-X e estende suas operações,

A arquitetura MIPS (acrônimo do inglês *Microprocessor without Interlocked Pipeline Stages*) foi desenvolvida originalmente na *Stanford University* [Patterson and Hennessy 2005]. Esta arquitetura, baseada no modelo *Harvard*, possui trinta e dois registradores de trinta e dois bits cada. A arquitetura possui três formatos de instruções básicas, denominadas de R (registrador), I (imediato) e J (jump ou salto). Nesta arquitetura o acesso à memória é feito sempre através de um dos registradores de trinta e dois bits.

O projeto BIP (acrônimo do inglês *Basic Instruction-set Processor*), vem sendo desenvolvido nos últimos doze anos [Vieira et al. 2016]. Ao longo de pouco mais de uma década, o projeto passou por diversas fases, versões e melhorias. A primeira versão do projeto é chamada de BIP I e possui uma arquitetura RISC (acrônimo do inglês *Reduced Instruction-Set Computer*). Para manter uma estrutura regular, simples e, além disso, menos complexa que as arquiteturas de tradicionais processadores ARM e MIPS, os seus autores optaram por uma arquitetura orientada à acumulador. A partir do BIP I surgiram outros modelos, como o μ BIP, BIP II, III e IV. Todos estes novos modelos são baseados nos atributos do modelo mais básico (BIP I) e estendem sua arquitetura com mais recursos, como novas instruções.

As seções a seguir descrevem a arquitetura do processador μ PD, proposto neste artigo como sendo um processador didático para uso em disciplinas de graduação em Arquitetura e Organização de Computadores e Processamento Digital de Sinais para os cursos de Ciência da Computação e Engenharia Elétrica.

3. Arquitetura do processador μ PD

A concepção do processador μ PD foi baseada nas seguintes premissas:

- Utilizar como referência uma arquitetura de processador de propósito geral bem conhecida no meio acadêmico e com literatura igualmente bem conhecida; e
- Implementar funções de processamento digital de sinais fora da estrutura da Unidade Lógica e Aritmética (ULA), isolando assim as funções de processamento de uso geral das funções de processamento digital de sinais.

Para atender a premissa de uso de uma arquitetura de processador bem conhecida, foram implementadas as instruções de máquina do processador MIPS. Assim, o processador μ PD utiliza mnemônicos similares aos do processador MIPS, de modo que os trinta e dois bits de cada instrução divididos em três tipos básicos: tipo R, tipo I e tipo J. A lista com os opcodes usados no processador μ PD é mostrada na Figura 1.

Foram criadas três instruções especiais para tratar dos blocos funcionais MAC, G711 *Encoder* e G711 *Decoder*. São instruções do tipo R e são mostradas na lista de mnemônicos e opcodes da Tabela 1. Os valores R[31:0] na tabela indicam que qualquer um dos trinta e dois registradores pode ser usado como registrador Rs, Rt e Rd. Os valores “xxxxx” indicam que os valores presentes naqueles campos são irrelevantes e, conseqüentemente, serão ignorados na execução da instrução. Os valores para os novos *opcodes* estão fora da faixa numérica coberta pelo processador, mostrado na Figura 1.

		Formato tipo R																																
		OPCODE					RS					RT					RD					SHIFT (SHAMT)				FUNCTION								
Classe	Mnemônico	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Instruções de Aritméticas e Lógica	add	0	0	0	0	0	0																											
	addu	0	0	0	0	0	0																											
	and	0	0	0	0	0	0																											
	nor	0	0	0	0	0	0																											
	or	0	0	0	0	0	0																											
	xor	0	0	0	0	0	0																											
	sll	0	0	0	0	0	0																											
	srl	0	0	0	0	0	0																											
	sub	0	0	0	0	0	0																											
	subu	0	0	0	0	0	0																											
Instrução de desvio	jr	0	0	0	0	0	0																											
Instruções de comparação	slt	0	0	0	0	0	0																											
	sltu	0	0	0	0	0	0																											

		Formato tipo I																																
		OPCODE					RS					RT					IMMEDIATE																	
Instruções de Aritméticas e Lógica	addi	0	0	1	0	0	0																											
	addiu	0	0	1	0	0	1																											
	ori	0	0	1	1	0	1																											
Instruções de Transferência (load/store)	lbu	1	0	0	1	0	0																											
	lhu	1	0	0	1	0	1																											
	lui	0	0	1	1	1	1																											
	lw	1	0	0	0	1	1																											
	sb	1	0	1	0	0	0																											
	sh	1	0	1	0	0	1																											
	sw	1	0	1	0	1	1																											
Instruções de desvio	beq	0	0	0	1	0	0																											
	bne	0	0	0	1	0	1																											
Instruções de comparação	slti	0	0	1	0	1	0																											
	sltiu	0	0	1	0	1	1																											

		Formato tipo J																															
		OPCODE					ADDRESS																										
Instruções de desvio	j	0	0	0	0	1	0																										
	jal	0	0	0	0	1	1																										

Figure 1. Instruções dos tipos R, I e J do processador μ PD são similares às instruções do processador MIPS.

O objetivo foi não alterar as instruções tradicionais do processador MIPS apresentados naquela figura. Para cobrir esta escolha, foi elaborado um programa montador, escrito em linguagem JAVA, que gera o código binário para o processador didático.

Table 1. Operações e Operandos das funções MAC e G711

Mnemonic	Opcode	Rs	Rt	Rd	Shift (shamt)	Func
Encoder G711	111101	R[31:0]	R[31:0]	R[31:0]	xxxxxx	xxxxxx
Decoder G711	111110	R[31:0]	R[31:0]	R[31:0]	xxxxxx	xxxxxx
MAC	111111	R[31:0]	R[31:0]	R[31:0]	xxxxxx	xxxxxx

4. Organização do processador μ PD

No processador μ PD proposto neste artigo o ciclo de instrução pode ter até cinco etapas. A Figura 2 ilustra os ciclos de máquina possíveis para o processador μ PD. Como forma de minimizar o número de ciclos de clock para cada instrução, a arquitetura do processador μ PD realiza os estágios de *memory*, *write back*, e MAC em um mesmo ciclo de *clock*, visto que estas instruções não são executadas em um mesmo instante de tempo.

O processador μ PD é dividido em dois blocos básicos, denominados de “unidade de controle” e “unidade de dados”, conforme ilustra a Figura 3(a). A unidade de controle está relacionada à decodificação das instruções e a geração de sinais de controle necessários para realizar cada instrução de máquina. A unidade de dados está relacionada à execução das instruções em si.

Na arquitetura proposta para o processador μ PD, existem registradores específicos para o contador de programa (*Programm Counter* - PC) e o registrador de instrução, de-

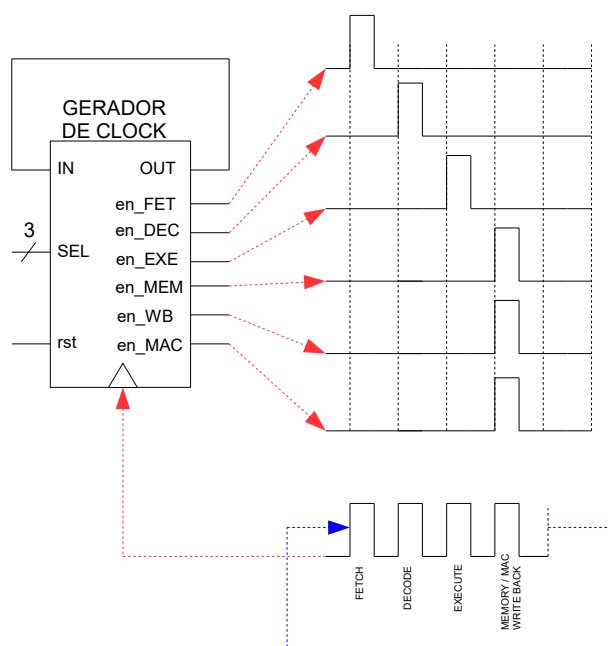


Figure 2. O processador μ PD é composto por até cinco ciclos de *clock*, sendo os três últimos realizados um a cada instrução.

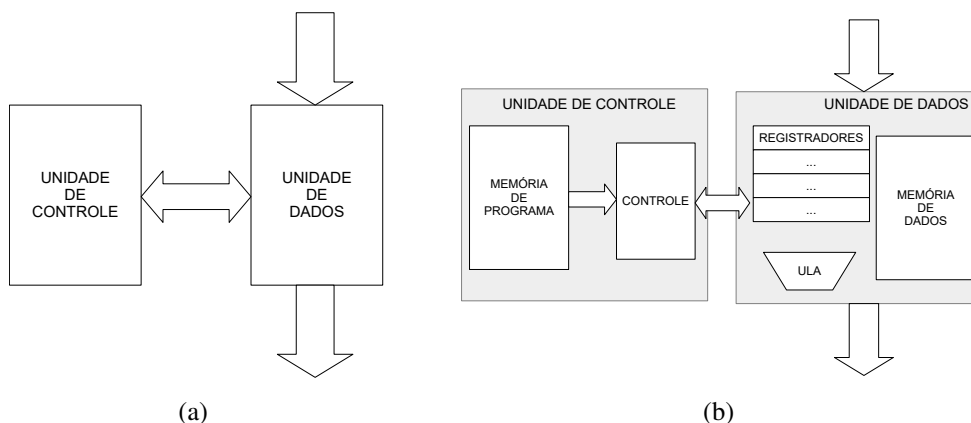


Figure 3. O processador μ PD dividido entre unidade de controle e unidade de dados.

nominado de IR (acrônimo de *Instruction Register*), além dos trinta e dois registradores de propósito geral. Uma divisão dos principais componentes do processador μ PD é mostrada na Figura 3(b). Note que os trinta e dois registradores estão alocados na unidade de dados. Os registradores PC e IR, não representados nesta figura, estão alocados na unidade de controle.

A Figura 4 apresenta os blocos internos que formam o processador μ PD. Uma borda quadrada, com indicação “ μ PD”, delimita os componentes do processador. Note que a unidade de dados e a unidade de controle não compartilham o mesmo barramento e, por este motivo, o μ PD é um processador baseado no modelo “*Harvard*”. A área em cor cinza claro contém os blocos funcionais da unidade de controle, enquanto que a área cinza escuro contém os componentes que formam a unidade de dados.

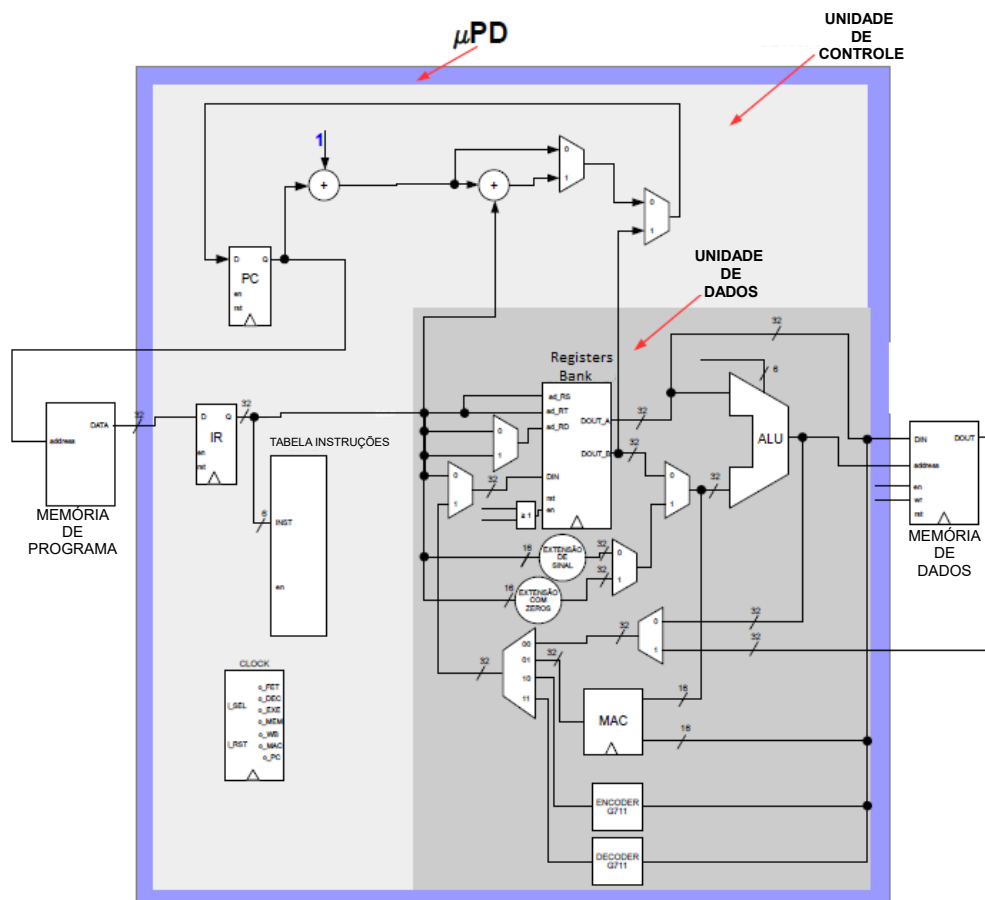


Figure 4. Diagrama em blocos do processador μ PD.

Como a proposta deste microprocessador é servir de ferramenta auxiliar no aprendizado de disciplinas de processamento digital de sinais, foram disponibilizados componentes na organização do μ PD voltados para o ensino de técnicas de processamento digital de sinais. Tais componentes estão localizados na parte baixa da Figura 4, na área cinza escuro: MAC (acrônimo de multiplicação e acumulação), *Encoder G711* e o *Decoder G711*. A operação de MAC exige um ciclo de *clock* para cada produto de dois termos e consequente acúmulo com o valor já existente no registrador de destino especificado pelo usuário. Por sua vez, as operações de *Encoder G711* e *Decoder G711* são implementadas na forma de circuitos combinacionais e não necessitam de ciclos de *clock* para apresentar os resultados de suas operações.

5. Resultados Experimentais

5.1. Consumo de recursos de um FPGA

Nesta subseção são apresentados os resultados relativos à avaliação que foi feita considerando o custo de silício na implementação do processador μ PD, quando sintetizado para a tecnologia FPGA. O processador foi sintetizado para um FPGA do fabricante *Xilinx*[®], denominado de XA7Q15TCPG236-2I. Os resultados foram obtidos em termos de *Flip-flop* (FF), *Block RAM* (BRAM) e *Lookup tables* (LUT), conforme mostra a Tabela 2. O consumo de LUTs ficou abaixo de 11% do total de LUTs disponíveis para

o FPGA utilizado na síntese. O uso de uma BRAM para ser utilizada como tabela de decodificação das instruções do μ PD foi importante para este baixo consumo de LUTs.

Table 2. Recursos do FPGA utilizado na síntese do μ PD.

Recurso	Estimativa	Disponível	Utilização %
FF	1111	20800	5,34
LUT	1111	10400	10,68
BRAM	1	25	2,00
DSP48	1	45	2,22

O codec G711 foi implementado como um bloco e lógica combinacional. O consumo de LUTs e FFs também se mostrou pouco significativo, frente ao total de recursos disponíveis no FPGA alvo, conforme mostra a Tabela 3. O total de LUTs e FFs foi menor do que 1% do total disponível.

Table 3. Recursos do FPGA utilizado na síntese do codec G711.

Recurso	Estimativa	Disponível	Utilização %
FF	41	20800	0,20
LUT	53	10400	0,51

5.2. Uso do μ PD em sala de aula

O processador μ PD foi implementado por alunos da quinta Fase do curso de Engenharia Elétrica, na disciplina de Microcontroladores e Microprocessadores. A concepção dos blocos funcionais, como caminho de dados, caminho de controle, MAC, etc., foi previamente definido, assim como a interconexão entre tais blocos funcionais. Desta forma, coube aos alunos a implementação interna dos blocos funcionais e sua validação através de simulações com arquivos de *testbenches*.

Um questionário com quatro perguntas foi aplicado aos alunos da disciplina de “Microcontroladores e Microprocessadores” em três semestres (dois anteriores ao uso do μ PD em sala de aula e um no semestre em que ele foi utilizado). A ideia era ter uma medida “qualitativa” sobre o uso do processador μ PD em sala de aula. A primeira pergunta feita no início de cada semestre letivo era: *Você possui conhecimento prévio de como funciona um processador?*. Em todos os semestres a resposta foi que 100% dos alunos desconheciam o funcionamento de um processador. O objetivo desta pergunta era saber se havia alunos que já cursaram, total ou parcialmente, alguma disciplina que tivesse apresentado noções sobre o funcionamento de um processador.

No início de cada semestre, os alunos desta disciplina conheceram os conceitos fundamentais sobre a arquitetura do processador MIPS 32, como modos de endereçamento, quantidade de registradores, etc.. Em seguida, foram realizados exercícios de laboratório com a implementação de códigos em linguagem *assembly*. Como ferramenta de montagem e simulação foi utilizado o simulador MARS, desenvolvido pela Universidade do Missouri (EUA). Após este período de aulas práticas de programação com linguagem de montagem, foi feita a segunda pergunta aos alunos: *Você entendeu plenamente como é o fluxo de dados entre memórias e registradores no processador MIPS?*

Do total de alunos apenas 65% deles disseram ter entendido plenamente como este fluxo de informações entre memórias e registradores acontecia no MIPS.

No terceiro semestre de avaliação foi iniciada a implementação do processador μ PD em linguagem de síntese (VHDL), depois das aulas de programação em linguagem assembly. Esta estratégia mostrou-se eficiente, pois os alunos puderam entender melhor a implementação pois tinham o conhecimento prévio de qual o comportamento esperado no processamento de cada instrução de máquina. As instruções implementadas pelos alunos durante o semestre foram: carregar um dado da memória em um registrador, realizar operações de soma e subtração, e enviar o resultado de uma operação aritmética de um registrador para a memória. A terceira pergunta feita aos alunos foi feita na última semana de aulas de cada semestre: *Você entendeu como o processador MIPS executa as instruções de um programa?*. Nos semestres que antecederam o uso do processador μ PD, 60% dos alunos disseram não entender como isso acontecia. Porém, no semestre em que foi utilizado o μ PD, 95% dos alunos disseram entender melhor como o processador executava as instruções de um programa.

Ainda no terceiro semestre foram trabalhadas as instruções específicas de processamento digital de sinais (MAC, *Encoder G711* e *Decoder G711*). A quarta pergunta, feita também na última semana de aula, foi: *Ficou claro para você como funcionam as instruções de DSP e qual sua utilização?*. Do total de alunos, 98% indicaram que compreenderam plenamente como tais instruções funcionavam e sua aplicabilidade em processamento digital de sinais. Neste ponto ressaltamos que não havia ferramenta similar para comparação de funções de especificadas neste projeto (no caso, a Instituição utilizava regularmente a ferramenta MARS).

6. Conclusão e Trabalhos futuros

A estratégia adotada em sala de aula com a implementação de um processador de 32 bits foi positiva e mostrou-se importante como agente motivador para o aprendizado dos conceitos de arquitetura e organização de processadores. A estratégia de oferecer a definição dos blocos funcionais previamente conectados facilitou o aprendizado dos alunos, além de propiciar a implementação do processador em poucas aulas semanais de laboratório. Entendemos que sem esta estratégia, a implementação poderia tomar um tempo superior ao disponível em um semestre letivo.

A implementação de outras instruções de máquina do μ PD continuará nos próximos semestres pelos futuros alunos da disciplina de Microcontroladores e Microprocessadores. A meta futura é implementar um compilador Assembly para o processador. Além disso, serão preparados experimentos específicos para serem utilizados na disciplina de Processamento Digital de Sinais. O bloco funcional de multiplicação e acúmulo é amplamente usado na implementação de filtros digitais de resposta ao impulso finita (FIR) e resposta ao impulso infinita (IIR). Além disso, o codec G711 implementado no processador μ PD é útil na realização de experimentos sobre compressão de dados (áudio) devido a sua simplicidade de implementação e entendimento.

7. Referências

References

- Clements, A. (1999). Selecting a processor for teaching computer architecture. *Microprocessors and Microsystems*, 23(5):281 – 290.
- Furber, S. (2000). *ARM System-on-Chip Architecture*. Addison-Wesley, 2nd edition.
- Nooshabadi, S. and Garside, J. (2005). Teaching embedded systems design - an international collaborative project. In *Proceedings Frontiers in Education 35th Annual Conference*, pages F2D–F2D.
- Patterson, D. A. and Hennessy, J. L. (2005). *Organization and architecture of computers*. Elsevier, 3ed edition.
- Vieira, P. V., Raabe, A., and Zeferino, C. A. (2016). Projeto bip: Impactos de 10 anos de uso de uma proposta interdisciplinar de ensino de computação. *International Journal of Computer Architecture Education (IJCAE)*, pages 32–37.
- Weber, R. F. (2012). *Fundamentos de Arquitetura de Computadores*. Bookman, 4ed edition.