

Universidade Católica Dom Bosco  
Renan Mira De Lima e Wilson Ricardo Martins Lopes Júnior  
Engenharia de Computação

## **ARM9**

Relatório apresentação para a avaliação  
do trabalho da disciplina de Arquitetura  
de Computadores II

Campo Grande – MS

Novembro/2007

## 1) O Que Foi Implementado?

Foram implementadas algumas instruções do tipo *DataProcessing*, tipo *System Interrupt* e do tipo *Branch*, sendo elas:

<u>Tipo</u>	<u>Instrução</u>	<u>Comando</u>
<i>DataProcessing</i>	ADD	RD:= Op1 + Op2
<i>DataProcessing</i>	AND	D:= Op1 AND Op2
<i>DataProcessing</i>	BIC	D:= Op1 AND NOT Op2
<i>DataProcessing</i>	EOR	RD:= Op1 XOR Op2
<i>DataProcessing</i>	ORR	RD:= Op1 + Op2
<i>DataProcessing</i>	RSB	RD:= Op2 – Op1
<i>DataProcessing</i>	SUB	RD:= Op1 - Op2
<i>DataProcessing</i>	CMP	Set cc on R1 – R2
<i>DataProcessing</i>	CMN	Set cc on R1 + R2
<i>DataProcessing</i>	TST	Set cc on R1 AND R2
<i>DataProcessing</i>	TEQ	Set cc on R1 XOR R2
<i>DataProcessing</i>	MVN	RD:= NOT R1
<i>DataProcessing</i>	MOV	RD:=R1
<i>Branch</i>	B	B := #Constante
<i>System Interrupt</i>	NOP	Espaço Vazio
<i>System Interrupt</i>	SYS_CALL	Interrupção Do Processo

As instruções estão no formato que o grupo conseguiu assimilar do arm9. Os códigos e conteúdos estão descritos nos arquivos enviados em conjunto.

## 2) O que Faltou Implementar?

Dentro dos arquivos gerado e do conjunto de instruções, não foi implementada as instruções e parte de instruções que utilizam o registrador de estado para operações especiais de add, sub e derivadas. Dentre os tipos de instruções, as do tipo *Multiply* e *Load and Store*, não foram implementamos. Tal fato se deve à falta de tempo decorrente por parte do grupo e por dificuldade de entender o formato de tais instruções.

### 3) Sintaxe:

<u>Ação Desejada</u>	<u>Comando Decorrente</u>
Geração Dos Arquivos Do Compilador	<code>\$/../bin/acsim arm9.ac -dy</code>
Realização Da Compilação	<code>\$ make -f Makefile.archc</code>
Execução Do Código	<code>\$ ./arm9.x --load=nome_arquivo.hex</code>

### 4) Bugs/Falhas Detectados e Não Corrigidos:

Durante a execução parte do que esta acontecendo será mostrado na tela de processamento e códigos do tipo *Multiply* e *Load and Store* não serão executadas e operações que utilizam carrays apresentaram erros.

### 5) Dificuldades Encontradas :

A maior dificuldade encontrada e vista, diz respeito à diferença entre a linguagem do ArchC para o MIPS contudo não se pode esquecer das dificuldades nas adaptações que tiveram q ser feitas nas instruções para que estas funcionem normalmente já que um ADD em MIPS fazia um único tipo de operação e em ArchC elas poderia fazer diversos tipos. Foi encontrado dificuldades na leitura e interpretação de conteúdos que traziam informação da arquitetura assim implementada, assim como compreensão do *DataSheet de cada instrução*.

No inicio do trabalho, foi apresentada grande dificuldade nos horários e adaptação para o mesmo já que um dos componentes do grupo trabalhava no período vespertino e outro no período noturno, problema que foi resolvido de maneira que os dois resolveriam o problema e estes discutiram em final de semana ou de manha em algum horário de aula. Outra dificuldade que pode ser notada é que algumas instruções são bem diferenciadas das da arquitetura MIPS e com isso seu entendimento ficou um pouco defasado, porém foi sanada esta duvida com um pouco de leitura e consulta os colegas e professor da disciplina. Em seguida vieram problemas nas próprias instruções onde tivemos que criar funções alternativas para que estes problemas fosse resolvidos como foi o caso da instrução nops e sys\_call assim como o uso de uma função auxiliar para que fosse feito 'rotate' em determinados casos de algumas operações que geravam 'overflow' que com um pouco e trabalho vieram a resolver o problema.

Foi assim tomado de partida a instruções de mais fácil compreensão as de 'Data Processing' e com elas foi tido um melhor aprendizado sobre a linguagem que teve um salto que nos levou ao desenvolvimento de

instruções do tipo 'System Interrupt', do tipo 'Branch' e do tipo 'Load/Store' porém esta última não foi concluída. Com o desenvolvimento do tipo 'Branch' foram feitas novas instruções do tipo 'Data Processing' que geravam saltos tais como CMP, CMN entre outras. Nosso objetivo sempre foi entender cada instrução sabendo o que cada uma fazia e em que parte do pipeline era realizada tal instrução.

## **6) Conclusão e Trabalhos Futuros**

Para os trabalhos futuros pretende-se chegar perto do Maximo ou continuar o seu desenvolvimento para melhor entendimento do arm9 e para uso em disciplinas futuras, com o intuito que este desenvolvimento sirva para sanar dúvidas sobre tal arquitetura e seu funcionamento e que sirva para outras disciplinas. Estima-se a conclusão de, pelo menos, duas ou três instruções de cada tipo sendo que as de Load e Store estavam em fase de término porém não foram postas por ainda apresentarem alguns erros de acesso e escrita.

Concluimos com isso que a arquitetura arm9 é uma estrutura de grande abrangência e de fato por ser tão ampla e as vezes complexa de se entender, sendo que com um pouco de esforço fica até fácil resolve-la. É uma arquitetura completa com um repertório enorme e amplo de instruções variando em seus tipos e funcionalidade. Como estudado anteriormente a estrutura MIPS este se comparado é mostrado de forma clara porém seu entendimento é um pouco mais complicado.

---