

Vinícius Jeremias dos Santos (GRR20232361)

Profº Marco A. Zanata Alves

Arquitetura de Computadores

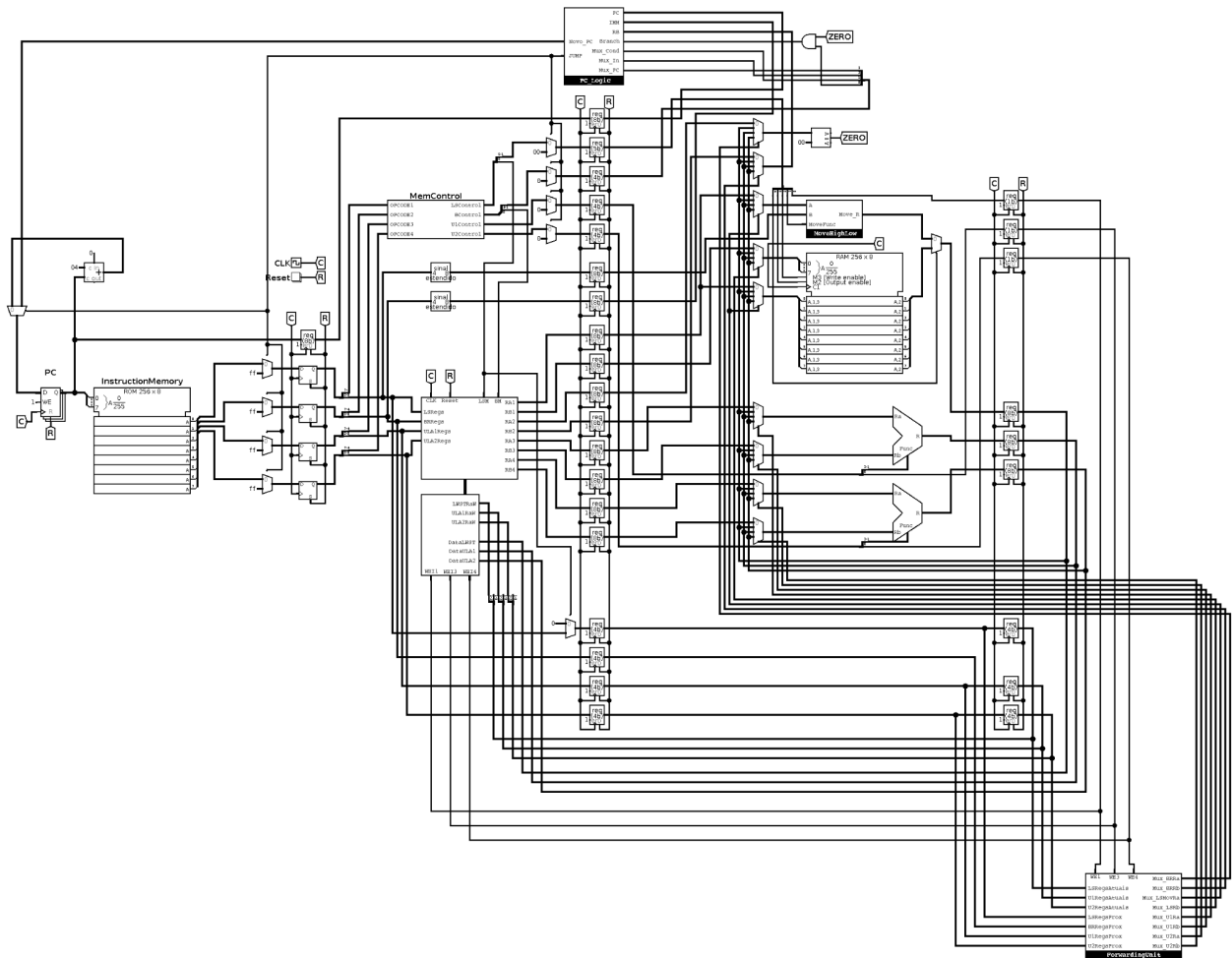
Trabalho - Arquitetura Sagui VLIW

Relatório de desenvolvimento do processador Sagui VLIW, um processador simplificado capaz de executar uma série de operações aritméticas para calcular uma soma de vetores de 12 posições e outros programas simples.

A nova versão do Sagui que segue os moldes de uma arquitetura VLIW contém uma memória endereçada a 4 bytes, ou seja, executa 4 instruções em paralelo que compartilham um mesmo banco de registradores e unidades funcionais diferentes. Definidas a princípio como LD/ST/MOV - BRANCH/JUMP - ULA 1 - ULA 2.

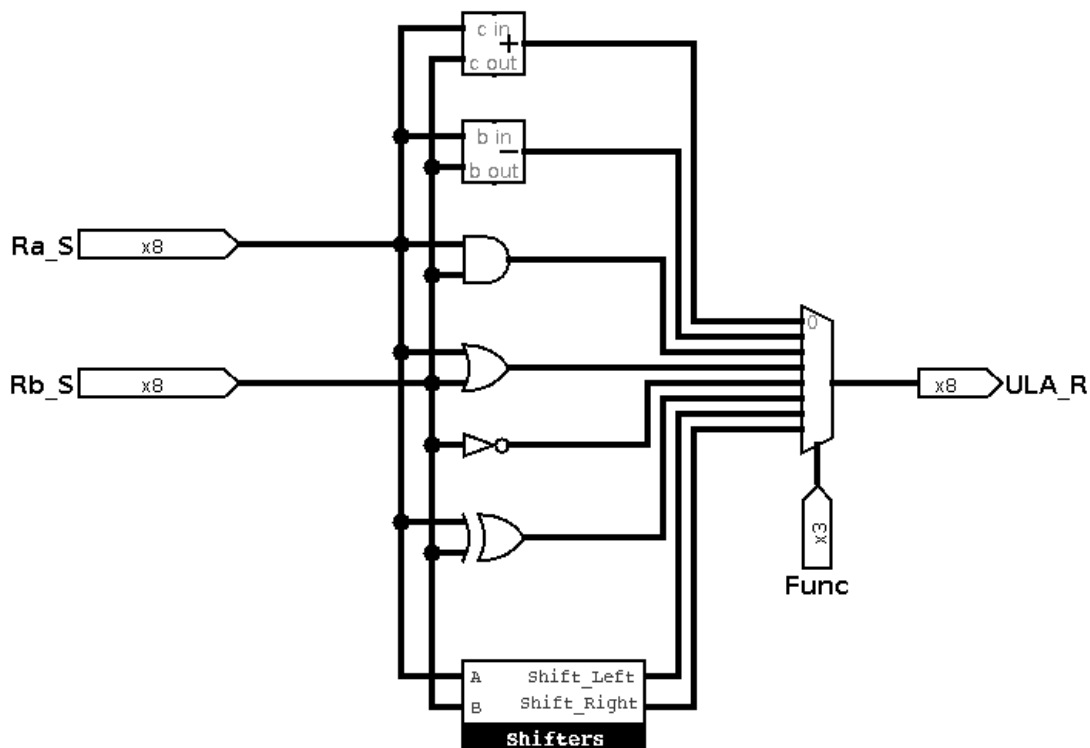
Além dos moldes VLIW, o Sagui foi implementado com Pipeline, permitindo uma maior vazão de instruções no processador, o que também requisitou uma série de adaptações nos componentes utilizados na primeira versão do Sagui, as principais mudanças estão no banco de registradores, que além de mais entradas de endereço para comportar a arquitetura VLIW também contém novas portas para endereços que foram propagados para estágios posteriores do pipeline, além de seus respectivos sinais de “Write Enable” e uma lógica combinacional para permitir a escrita correta nos registradores definidos na instrução.

Diagrama do Sagui



Como é observável neste diagrama, o processador possui alguns Mux antes das “Barreiras” do Pipeline, eles permitem que os dados corretos sejam propagados para os próximos estágios de execução das instruções, por exemplo, quando é necessário inserir “Nops” durante a execução do programa após um salto condicional ou incondicional, e também para propagar o registrador utilizado em instruções que utilizam do imediato e $R[0]$.

O Projeto da ULA



O projeto da ULA foi adaptado para o Sagui VLIW, executando um Xor ao invés da antiga unidade que executava o Move High/Low/Register, os Shifters também foram adaptados para funcionar corretamente, fora tais mudanças, prossegue executando operações mais simples. Devido às novas propriedades da arquitetura, o processador utiliza duas ULA 's em paralelo para executar as instruções lógico-aritméticas definidas por padrão em uma palavra grande.

Forwarding Unit

Para definir a prioridade dos dados em estágio de execução que devem ser adiantados, a Forwarding Unit utiliza de uma lógica similar ao do banco de registradores, que compara os endereços de escrita no banco de registradores e verifica se o valor será atualizado no próximo sinal de Clock. Após passar por essa lógica, a unidade emite sinais que são conectados a Mux no estágio de execução do processador, que selecionam os dados atualizados para a operação atual, permitindo que problemas em torno de dependências de controle sejam resolvidos pela própria arquitetura.

Sinais de Controle

BR/JUMP	Mux_PC	Mux_In	Branch	Mux_BR
brzr	0	0	1	1
brzi	0	0	1	0
jr	1	1	0	0
ji	1	0	0	0
nop	0	0	0	0

LW/ST/MOV	Mux_WB	Load	Store	MoveH/L	WE	Mux_LS
ld	1	1	0	0	1	1
st	0	0	1	0	0	1
movh	0	0	0	1	1	0
movl	0	0	0	0	1	0
nop	0	0	0	0	0	0

ULA 1/2	Func	WE
add	000	1
sub	001	1
and	010	1
or	011	1
slr	100	1
srr	110	1
nop	000	0

Para os sinais de controle, as instruções foram separadas de acordo com seus respectivos OP-CODE's, desta forma, a memória de controle emite apenas os sinais de controle específicos para o funcionamento daquela instrução, diminuindo consideravelmente o número de bits de controle necessários para a execução de apenas uma única palavra (4 instruções).