



Documento de Micro arquitetura

Processador BLA

Efficiency Eletrronics S.A.

Build 2.0

History Review

Data	Descrição	Autor(es)
19/02/2016	Descrição dos blocos funcionais do processador, incluindo entradas e saídas	Thiago Sampaio Lima
20/02/2016	Revisão dos blocos funcionais e adição dos componentes restantes	Matheus Moura Batista
21/02/2016	Definição do propósito do documento, conteúdo das seções (introdução) e organização do documento	Joacy Mesquita
03/03/2016	Revisão geral do documento	Todos
01/04/2016	Revisão geral e correção do documento	Todos

SUMÁRIO

1	Introdução	4
1.1	Propósito	4
1.2	Organização do documento	4
2	Características gerais	4
3	Diagrama de Blocos do Processador	4
4	Módulos	6
4.1	PC	6
4.1.1	Entradas	6
4.1.2	Saídas	6
4.2	Somador-1	6
4.2.1	Entradas	6
4.2.2	Saídas	6
4.3	Memória de Instrução	6
4.3.1	Entradas	6
4.3.2	Saídas	6
4.4	Unidade de Extensão de Sinal	7
4.4.1	Entradas	7
4.4.2	Saídas	7
4.5	Unidade de Extensão de Endereço	7
4.5.1	Entradas	7
4.5.2	Saídas	7
4.6	Unidade de Controle	7

4.6.1	Entradas	7
4.6.2	Saídas	7
4.7	Comparador	8
4.7.1	Entradas	8
4.7.2	Saídas	8
4.8	Banco de Registradores	8
4.8.1	Entradas	8
4.8.2	Saídas	9
4.9	Unidade de Detecção de Conflitos	9
4.9.1	Entradas	9
4.9.2	Saídas	9
4.10	Unidade Lógica e Aritmética	10
4.10.1	Entradas	10
4.10.2	Saídas	10
4.11	Memória de Dados	10
4.11.1	Entradas	10
4.11.2	Saídas	11

1. Introdução

1.1. Propósito

O propósito deste documento é apresentar uma visão detalhada da microarquitetura do processador BLA, incluindo suas principais características e a descrição dos seus componentes.

1.2. Organização do documento

As seções deste documento são organizadas conforme estruturado abaixo.

- Seção 2: Apresenta as principais características da microarquitetura incluindo questões como CPI e organização do pipeline.
- Seção 3: Apresenta uma visão geral da microarquitetura do processador, apresentando todos os módulos funcionais do sistema e suas conexões.
- Seção 4: Apresenta o detalhamento dos blocos individuais do sistema, com indicação dos sinais de entrada e saída e seu propósito.

2. Características gerais

A microarquitetura foi desenvolvida para executar instruções em três estágios funcionais. O primeiro estágio é responsável por buscar e decodificar a instrução e, adicionalmente, realizar a busca dos operandos. Ainda no primeiro estágio são feitas as decisões de salto e verificação de conflitos de dados. Já no segundo estágio são feitas todas as operações lógicas e aritméticas (ULA) e as operações de acesso à memória. No terceiro e último estágio é realizada a gravação dos dados nos registradores.

Devido a forma que a microarquitetura foi desenvolvida, sem conflitos de controle e sem bolhas no pipeline, a execução em pipeline possui um CPI (ciclos por instrução) de 1,0.

No que diz respeito à unidade de controle, esta foi descrita com o metodologia *hardwire*, que tem como principal vantagem a maior velocidade na decodificação de instruções. Sua principal desvantagem está na complexidade gerada no módulo. Porém, devido ao pequeno número de instruções utilizadas, a complexidade apresentada se torna satisfatória.

3. Diagrama de Blocos do Processador



4. Módulos

4.1. PC

O contador de programa (PC) recebe como entrada o endereço da próxima instrução a ser buscada e armazena este endereço até que um sinal de clock seja detectado. Quando um sinal de reset é recebido o endereço do PC é reinicializado.

4.1.1. Entradas

- **Endereço:** Endereço da próxima instrução que será executada.
- **Clock:** Sinal de clock do processador utilizado para controlar o tempo de atualização do PC.
- **Reset:** Sinal utilizado para zerar o endereço contido no PC.

4.1.2. Saídas

- **Endereço:** Endereço da instrução que deverá ser buscada na memória de instruções.

4.2. Somador-1

O somador-1 recebe o endereço armazenado no contador de programa e o adiciona com 1 para determinar qual o endereço da próxima instrução a ser buscada. O endereço resultante é enviado para a saída do bloco e retorna ao contador de programa caso a instrução anterior não tenha sido um salto ou a condição para o salto não tenha sido atendida.

4.2.1. Entradas

- **Endereço** Endereço da instrução em execução no processador.

4.2.2. Saídas

- **Endereço:** Endereço da próxima instrução que será buscada (endereço de entrada + 1).

4.3. Memória de Instrução

A memória de instrução recebe como entrada o endereço da próxima instrução a ser executada. Após buscar a instrução correspondente, a mesma é colocada na saída do bloco.

4.3.1. Entradas

- **Endereço:** Endereço recebido do contador de programa (PC) para que seja buscada uma nova instrução a ser executada.

4.3.2. Saídas

- **Instrução:** Instrução buscada na memória

4.4. Unidade de Extensão de Sinal

A Unidade de extensão de sinal recebe como entrada uma constante de 16 bits e a estende para 32 bits, colocando-a na saída posteriormente. Esta extensão é feita replicando o bit mais significativo da constante a todos os bits a esquerda do mesmo.

4.4.1. Entradas

- **Dados:** Constante com 16 bits.

4.4.2. Saídas

- **Dados:** Constante estendida com 32 bits.

4.5. Unidade de Extensão de Endereço

A Unidade de extensão de endereço recebe como entrada um endereço de 12 bits e o estende para 32 bits, colocando-o na saída posteriormente. Esta extensão é feita preenchendo com zero todos os bits a esquerda do bit mais significativo do mesmo.

4.5.1. Entradas

- **Dados:** Endereço com 12 bits.

4.5.2. Saídas

- **Dados:** Endereço estendido com 32 bits.

4.6. Unidade de Controle

A unidade de controle recebe o tipo e código da operação fornecidos pela memória de instruções e os decodifica para determinar o que cada bloco do processador irá fazer com os dados que serão recebidos.

4.6.1. Entradas

- **Tipo e código de operação:** Tipo e código de operação fornecidos pela instrução para serem decodificados.

4.6.2. Saídas

- **Unidade lógica e aritmética:** Define o que a unidade lógica e aritmética deverá fazer com os dados de entrada.
- **Banco de registradores:** Determina o que deve ser feito com os dados de escrita recebidos pelo banco de registradores.
- **Memória de dados:** Define se a operação será de leitura ou de escrita de dados.
- **Comparador:** Define se o comparador deverá testar se a condição é verdadeira ou se é falsa.
- **Multiplexadores:** Determina aos multiplexadores qual dos barramentos de dados de entrada devem ser redirecionados para suas saídas.
- **Write Instruction:** Define se a instrução escreve ou não em um registrador.

4.7. Comparador

O comparador recebe como entradas uma condição de salto, as flags ativadas e desativadas e um sinal da unidade de controle determinando se o salto é do tipo “verdadeiro” ou “falso”. Após receber as entradas o comparador verifica se a condição de salto condiz com as flags presentes e então manda para a saída a ordem de realizar o salto ou não.

4.7.1. Entradas

- **Condição:** Código da flag a ser verificada para determinar se um salto deverá ou não ocorrer.
- **Flags:** Flags ativadas e desativadas a serem comparadas com a condição recebida.
- **Sinal da Unidade de Controle:** Determina se o salto condicional é do tipo salto-verdadeiro ou salto-falso.

4.7.2. Saídas

- **Salto:** Determina se o salto deverá ou não ser realizado.

4.8. Banco de Registradores

O banco de registradores identifica os registradores que serão utilizados através dos endereços recebidos e transfere os dados contidos neles para as respectivas saídas. Além disso, quando uma subida de clock é identificada e o sinal de gravação está ativo, os dados recebidos serão armazenados.

4.8.1. Entradas

- **Registrador A:** Endereço do registrador que representará o registrador A da operação.
- **Registrador B:** Endereço do registrador que representará o registrador B da operação.
- **Registrador C:** Endereço do registrador que representará o registrador C da operação.
- **Dados:** Dados que serão gravados no registrador de destino (registrador C). Estes dados podem conter o resultado de uma operação lógica e aritmética da ULA ou uma constante a ser carregada.
- **Endereço da próxima instrução (PC):** Endereço que será salvo no registrador r15 quando for uma instrução “jal”.
- **Sinal da Unidade de Controle:** O sinal de controle, neste contexto, atua indicando ao banco de registradores o que deve ser feito com os bytes recebidos através da entrada de dados e do PC. As operações disponíveis são: 1) escrever os dados no registrador C (bits ‘000’); 2) escrever os 2 bytes menos significativos dos dados nos bytes menos significativos do registrador C (bits ‘001’); 3) escrever os 2 bytes

menos significativos dos dados nos bytes mais significativos do registrador C (bits '010'); 4) Gravar o endereço do PC no registrador r15 (bits '011'); 5) Não fazer nada (bits '1XX').

- **Clock:** Sinal de clock do processador utilizado para controlar o tempo de gravação dos dados no registrador.

4.8.2. Saídas

- **Registrador A:** Dados armazenados no registrador A.
- **Registrador B:** Dados armazenados no registrador B.

4.9. Unidade de Detecção de Conflitos

A Unidade de detecção de conflitos verifica se há conflitos de dados no pipeline. Para isso é verificado se os registradores de leitura da instrução no primeiro estágio (registrador A e B) são iguais ao registrador de destino da instrução no estágio dois ou três. Se um conflito for identificado, a unidade realiza um forwarding através do envio de um sinal para a seleção de dados nos multiplexadores A e B.

4.9.1. Entradas

- **Código da operação:** Bits 22 e 23 da instrução. Estes bits são utilizados para determinar se a instrução realiza uma operação com constantes.
- **Registrador A:** Endereço do registrador A da instrução que está no primeiro estágio do pipeline.
- **Registrador B:** Endereço do registrador B da instrução que está no primeiro estágio do pipeline.
- **Write Instruction 1:** Determina se a instrução que está no segundo estágio do pipeline escreve em um registrador.
- **Registrador de Destino 1:** Endereço do registrador de destino da instrução que está no segundo estágio do pipeline.
- **Write Instruction 2:** Determina se a instrução que está no terceiro estágio do pipeline escreve em um registrador.
- **Registrador de Destino 2:** Endereço do registrador de destino da instrução que está no terceiro estágio do pipeline.

4.9.2. Saídas

- **Multiplexador A:** Sinal de controle do multiplexador A.
- **Multiplexador B:** Sinal de controle do multiplexador B.

4.10. Unidade Lógica e Aritmética

A Unidade lógica recebe como entradas dois operandos e um código de operação que é decodificado a fim de definir o funcionamento da mesma. Dependendo da operação recebida, a unidade lógica e aritmética pode utilizar um operando (operando 1) ou dois para realizar a operação em questão. Após a realização o resultado é colocado na saída e é verificada a existência ou não de flags para serem atualizadas.

4.10.1. Entradas

- **Operando 1:** Dado presente no registrador A especificado na instrução. Este dado é utilizado juntamente com o operando 2 ou sozinho (em algumas operações específicas).
- **Operando 2:** Dado presente no registrador B especificado na instrução. Este dado é utilizado juntamente com o operando 1.
- **Sinal da Unidade de Controle:** Código da operação especificado na instrução. Este código define o que a unidade lógica e aritmética irá fazer com os dados recebidos.

4.10.2. Saídas

- **Resultados:** Saída principal da unidade lógica e aritmética. Esta saída apresenta os resultados da operação em um barramento de 32 bits.
- **Flags:** Saída que indica as condições (flags) do resultado gerado na operação. São estas: resultado negativo, resultado zerado, resultado negativo ou zerado, resultado diferente de zero, carry, e overflow. Caso a operação não gere flags, a flag presente será mantida.

4.11. Memória de Dados

A memória de dados recebe como entradas o endereço onde a memória será acessada, dados (no caso de operação de escrita) e um sinal da unidade de controle que determina qual das duas operações (leitura ou escrita) será realizada. No caso de uma instrução de leitura de memória, o dado será buscado no endereço fornecido e colocado na saída de dados da memória. Já no caso de uma operação de escrita, os dados recebidos serão gravados no endereço fornecido pela instrução.

4.11.1. Entradas

- **Endereço:** Endereço de memória a ser acessado para leitura ou escrita de dados.
- **Dados:** Dados a serem escritos na memória.
- **Sinal da Unidade de Controle:** Sinal que indica a operação de memória que será realizada: leitura da memória a partir de um endereço especificado ou escrita de um dado num endereço de memória.
- **Clock:** Sinal de clock do processador utilizado para sincronizar o tempo de gravação dos dados na memória.

4.11.2. Saídas

- **Dados:** Saída dos dados obtidos em uma operação de leitura de memória.