



**Documento de Micro arquitetura**

Processador BLA

Efficiency Eletrronics S.A.

**Build 1.0**

## 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

## SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>4</b>
1.1	Propósito . . . . .	4
1.2	Organização do documento . . . . .	4
<b>2</b>	<b>Diagrama de Blocos do Processador</b>	<b>4</b>
<b>3</b>	<b>Módulos</b>	<b>6</b>
3.1	Unidade Lógica e Aritmética . . . . .	6
3.1.1	Entradas . . . . .	6
3.1.2	Saídas . . . . .	6
3.2	Banco de Registradores . . . . .	6
3.2.1	Entradas . . . . .	6
3.2.2	Saídas . . . . .	7
3.3	Memória de Dados . . . . .	7
3.3.1	Entradas . . . . .	7
3.3.2	Saídas . . . . .	8
3.4	Registrador de Flags . . . . .	8
3.4.1	Entradas . . . . .	8
3.4.2	Saídas . . . . .	8
3.5	Memória de Instrução . . . . .	8
3.5.1	Entradas . . . . .	8
3.5.2	Saídas . . . . .	8
3.6	PC . . . . .	8
3.6.1	Entradas . . . . .	9
3.6.2	Saídas . . . . .	9

3.7	Unidade de Extensão de Sinal . . . . .	9
3.7.1	Entradas . . . . .	9
3.7.2	Saídas . . . . .	9
3.8	Unidade de Extensão de Endereço . . . . .	9
3.8.1	Entradas . . . . .	9
3.8.2	Saídas . . . . .	9
3.9	Comparador . . . . .	9
3.9.1	Entradas . . . . .	9
3.9.2	Saídas . . . . .	10
3.10	Multiplexadores . . . . .	10
3.10.1	Entradas . . . . .	10
3.10.2	Saídas . . . . .	10
3.11	Somador-1 . . . . .	10
3.11.1	Entradas . . . . .	10
3.11.2	Saídas . . . . .	10
3.12	Unidade de Controle . . . . .	10
3.12.1	Entradas . . . . .	10
3.12.2	Saídas . . . . .	11

## **1. Introdução**

### **1.1. Propósito**

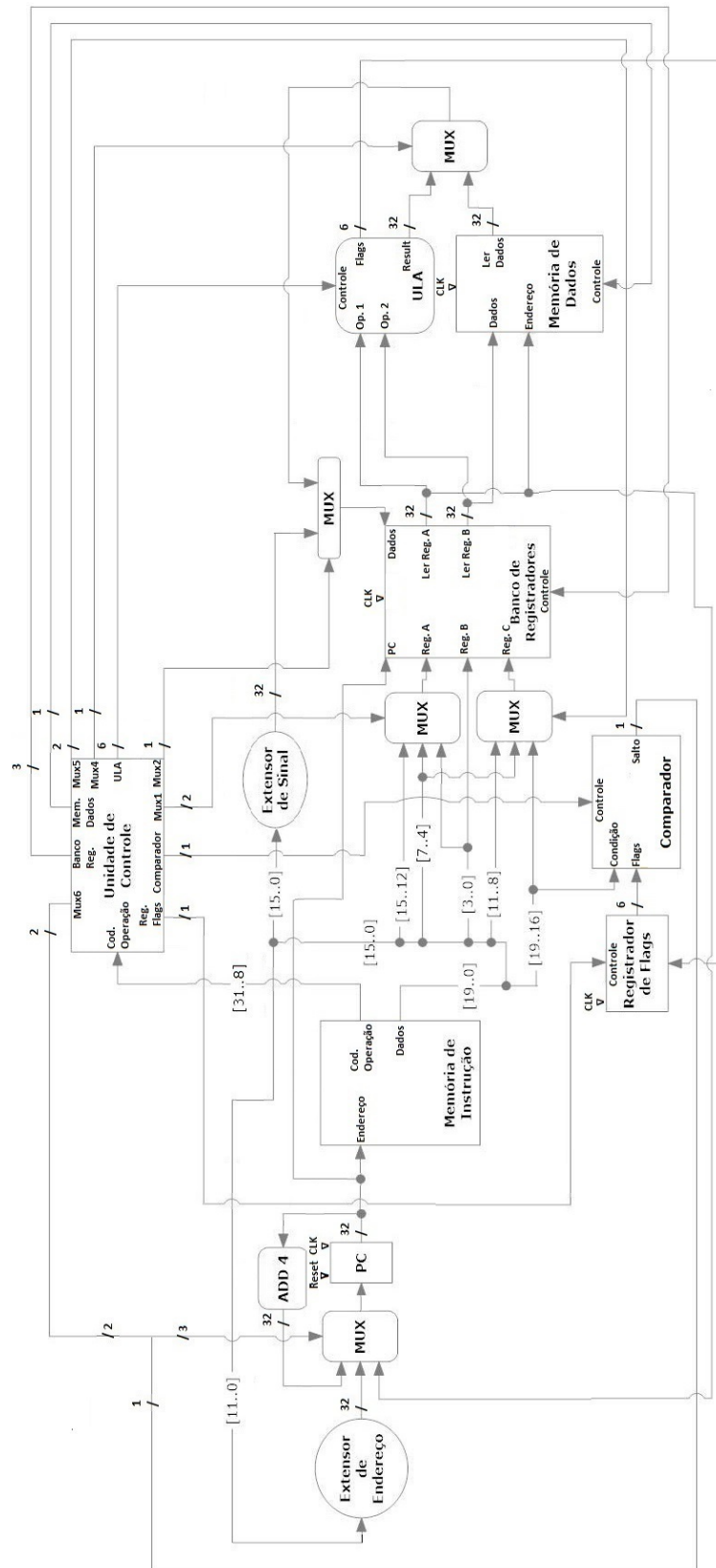
O propósito deste documento é mostrar uma visão detalhada da microarquitura do processador BLA, incluindo suas principais características e a descrição dos seus componentes.

### **1.2. Organização do documento**

As seções deste documento são organizadas conforme mostrado a seguir.

- Seção 2: Apresenta uma visão geral da microarquitura do processador, apresentando todos os módulos funcionais do sistema e suas conexões.
- Seção 3: Apresenta o detalhamento dos blocos individuais do sistema, com indicação dos sinais de entrada e saída e seu propósito.

## **2. Diagrama de Blocos do Processador**



**Figura 1: Microarquitetura do Processador BLA**

### 3. Módulos

#### 3.1. 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.

##### 3.1.1. 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.

##### 3.1.2. 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.

#### 3.2. 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. Essa gravação pode ser feita de diversas formas dependendo dos sinais recebidos na unidade de controle. Entre as formas possíveis estão: gravar dados nos bytes menos significativos ou mais significativos do registrador indicado ou o endereço de retorno no registrador r15.

##### 3.2.1. 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.

### 3.2.2. Saídas

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

## 3.3. 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.

### 3.3.1. 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 num determinado endereço de memória de um determinado dado.
- **Clock:** Sinal de clock do processador utilizado para controlar o tempo de gravação dos dados na memória.



### 3.3.2. Saídas

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

## 3.4. Registrador de Flags

O Registrador de flags recebe as flags geradas na unidade lógica e aritmética e armazena estes dados para posterior uso. Uma instrução de salto condicional, por exemplo, utiliza estas flags para determinar se o salto de fato acontecerá ou não. As flags são alteradas somente após uma operação lógica e aritmética.

### 3.4.1. Entradas

- **Flags da ULA:** Flags geradas durante as operações da unidade lógica e aritmética. No total são 6 flags, cada uma de 1 bit.
- **Sinal de Controle:** Indica se deve ou não atualizar as flags.
- **Clock:** Sinal de clock do processador utilizado para controlar o tempo de gravação das flags no registrador.

### 3.4.2. Saídas

- **Flags:** Flags atualizadas que podem ser utilizadas em condições para saltos.

## 3.5. Memória de Instrução

A memória de instrução recebe como entrada o endereço da próxima instrução a ser executada. Após buscar a instrução correspondente, é colocado nas saídas a parte da instrução que contém os dados e a que contém o tipo e código da operação.

### 3.5.1. Entradas

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

### 3.5.2. Saídas

- **Tipo e código da operação:** Indica o tipo (de salto, lógicas e aritméticas, de acesso à memória e de carregamento de constantes) da operação e o código da mesma. Este trecho da instrução vai para a unidade de controle.
- **Dados:** Dados contidos na instrução buscada. Estes dados podem incluir constantes, registradores, endereços de memória, condições de salto e endereços de salto.

## 3.6. 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 zerado.

#### 3.6.1. 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.

#### 3.6.2. Saídas

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

### 3.7. 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.

#### 3.7.1. Entradas

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

#### 3.7.2. Saídas

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

### 3.8. 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.

#### 3.8.1. Entradas

- **Dados:** Endereço com 12 bits.

#### 3.8.2. Saídas

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

### 3.9. 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.

#### 3.9.1. 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.

### 3.9.2. Saídas

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

## 3.10. Multiplexadores

Os multiplexadores recebem dois ou três barramentos de 4 ou 32 bits e, de acordo com o sinal recebido pela unidade de controle, determinam qual das entradas será redirecionada para a saída.

### 3.10.1. Entradas

- **Barramentos:** Dois ou três barramentos de 4 ou 32 bits a serem selecionados para a saída do multiplexador.
- **Sinal da Unidade de Controle:** Sinal que determina qual das entradas será redirecionada para a saída.

### 3.10.2. Saídas

- **Saída:** O Barramento selecionado pelo multiplexador.

## 3.11. 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.

### 3.11.1. Entradas

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

### 3.11.2. Saídas

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

## 3.12. 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.

### 3.12.1. Entradas

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

### 3.12.2. 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.
- **Registrador de flags:** Define se o registrador de flags deverá ou não sobrescrever as flags já armazenadas com os dados da entrada.
- **Multiplexadores:** Determina aos multiplexadores qual dos barramentos de dados de entrada devem ser redirecionados para suas saídas.