

# Laboratório 9

## Uso da Memória RAM com LPM

### Introdução

O objetivo deste experimento é aprender a usar parte das diversas memórias RAM que compõem o kit DE2-150. Para tanto será inicialmente adotada a biblioteca LPM em uma configuração simples, usando os blocos de memória RAM chamados de M9K, que são internos a todos os modelos da família Cyclone IV. Há outros módulos de memória, externas ao FPGA, no kit DE2-150: SSRAM (2 MB) e SDRAM (64 MB), além de um módulo flash de 8 MB e um slot para cartões SD. Neste momento, apenas será usada a M9K, deixando as demais para uma outra etapa.

Os módulos M9K são blocos de memórias RAM de 9 kbits, que compreendem 8192 bits de memória e mais 1024 bits de paridade, que podem ser usadas em diversos arranjos, como 8k x 1, 4k x 2, 2k x 4, 1k x 8, 1k x 9, 512 x 16, 512 x 18, 256 x 32 e 256 x 36. O chip EP4CGX150 possui um total de 6480 kbits de memória embarcada. A Figura 1 apresenta uma configuração que pode ser usada, usando apenas uma parte de um bloco M9K. Pode-se notar que há 5 linhas de endereçamento, unidirecionais, 8 linhas para o dado, bidirecionais, e o sinal de controle que determina se vai ocorrer um processo de escrita ou de leitura da memória.

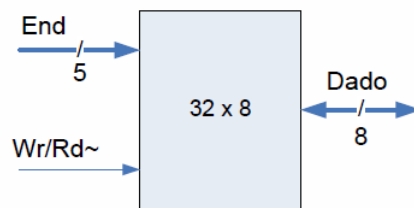


Figura 1: Configuração da memória a ser implementada.

Deve-se mencionar que os blocos M9K possuem registros que permitem sincronizar os processos de escrita e leitura, além de registradores separados para os dados que estão sendo escritos ou lidos. Assim, o circuito da Figura 2 apresenta maior detalhamento para o módulo de memória 32 x 8, e está de acordo com a forma como ele pode ser representado através do uso da biblioteca LPM.

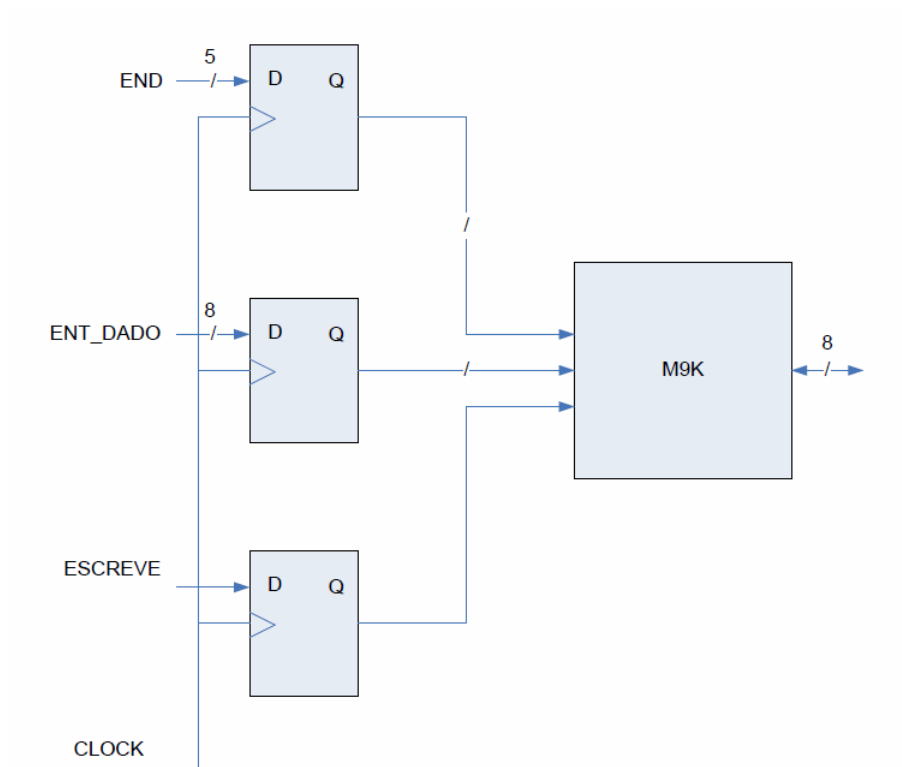


Figura 2: Circuito de memória RAM a ser implementado.

Observe no circuito da Figura 2 que a entrada de dados passa por um registrador, mas a saída de dados vem diretamente da memória, ambos com 8 bits. O endereçamento tem um registrador de 5 bits, e há ainda um sinal de controle da memória, também registrado e um sinal de relógio comum a todos os registradores, que sincroniza a operação da memória, tanto para ler como para escrever. Para as portas de entradas, os registradores são sempre obrigatoriamente usados, mas são opcionais para as portas de saída.

Para implementar o circuito da Figura 2, usando o módulo M9K disponível, note na Figura 2 que o barramento de dados é bi-direcional. Inicie o MegaWizzard Plug-in Manager, selecione a linguagem VHDL e atribua o nome `ramlpm` ao arquivo. Selecione a memória RAM dentro do conjunto Memory Compiler, como a RAM: 1-PORT, conforme pode ser visto na Figura 3.

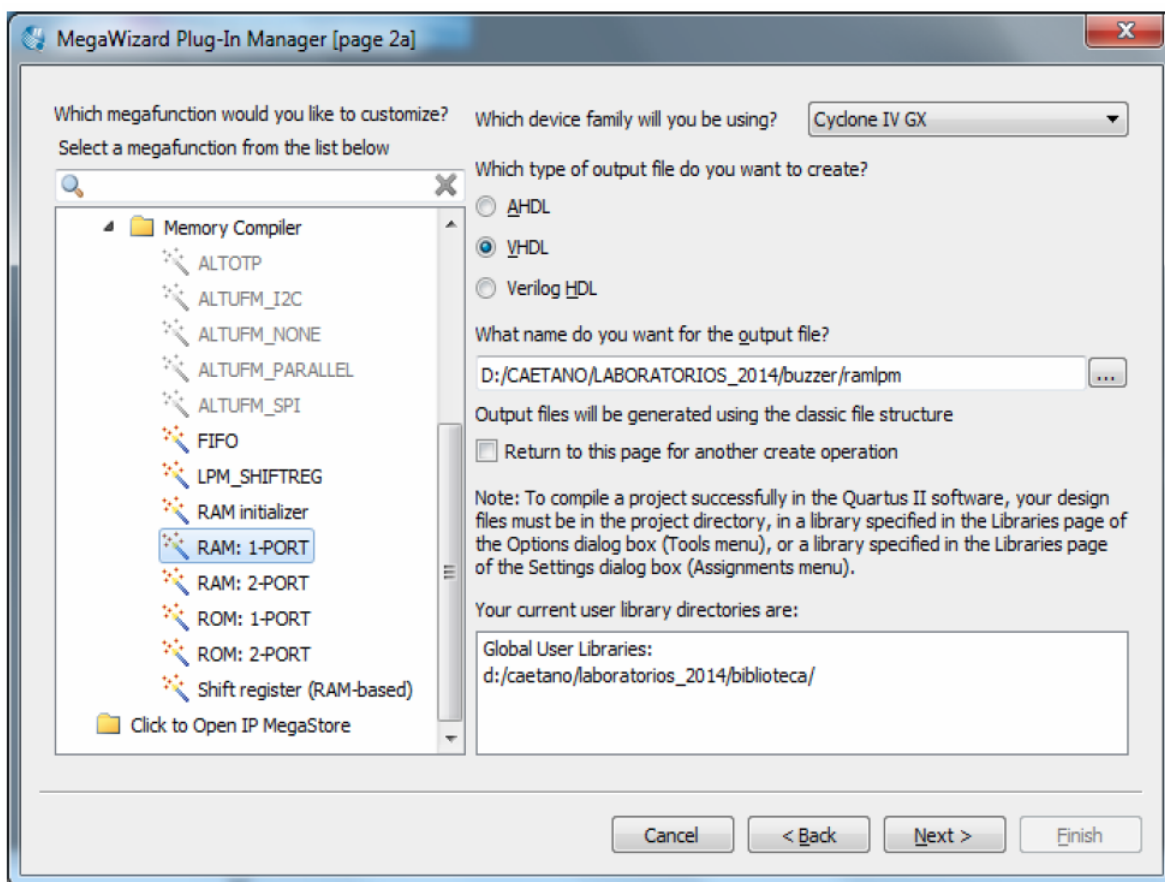


Figura 3: Seleção da memória M9K.

Na tela que vem a seguir, determine a configuração da M9K como 32 x 8 bits, com um único relógio. Continuando, deve surgir a tela apresentada na Figura 4. Nesta tela, elimine a marca que seleciona a saída registrada, de forma a manter apenas as demais entradas através de registradores, conforme a configuração determinada na Figura 2.

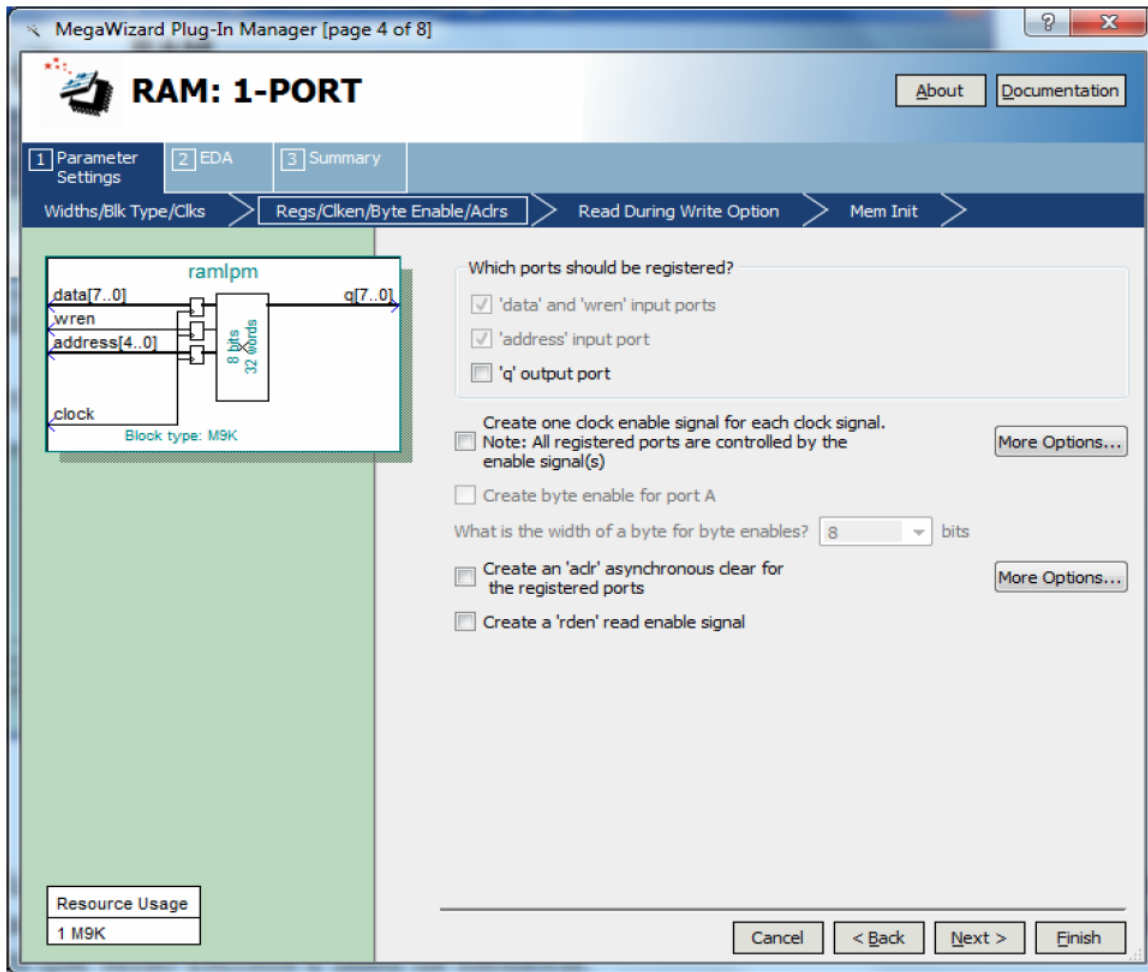


Figura 4: Tela de configuração dos registradores.

Para finalizar esta parte inicial de uso da memória, o código respectivo deve incluir portas ou sinais apropriados para todos os elementos do circuito em sua entidade principal, que permita determinar valores que serão escritos e lidos da memória.

Nos experimentos que fazem uso da memória acima, deverá ser utilizado a estrutura compreendendo registradores e unidade aritmética, vista anteriormente porém que será padronizada para evitar dificuldades. Assim, para a implementação desta estrutura, as seguintes entidades devem ser usadas como componentes, ocasionando um código principal com várias instâncias que resulta bastante simples e razoavelmente unificado para todos os grupos de alunos.

Notar que nos experimentos será implementada uma memória de 16 bytes e não de 32, devido ao número de chaves de entrada de dados.

## Parte I

Crie um novo projeto e implemente a configuração da Figura 5, usando a biblioteca LPM para criar a memória RAM com 16 endereços e dados de 8 bits. Use as chaves para entrada de dados, para entrada de endereço, e para o sinal de escrita/leitura, de acordo com a Tabela 1. Para o processo de escrita, mostre o dado a ser armazenado nos mostradores HEX(1-0), o endereço em HEX(7-6) e o sinal de escrita no LEDG(0). No processo de leitura,

CLKZ – KEY(0)	CLKM – KEY(1)	CLKR – KEY(2)	ClearRegs – KEY(3)
TecD - SW(7-0)	TecE – SW(11-8)	WrEn – SW(12)	SelM2- SW(13)
SelM1 - SW(14)	SelRB- SW(15)	SelRA - SW(16)	AddSubR – SW(17)

Tabela 1: Chaves para entrada e saída de dados

mostre o dado lido da memória nos HEX(1-0). Mostre o resultado aos professores.

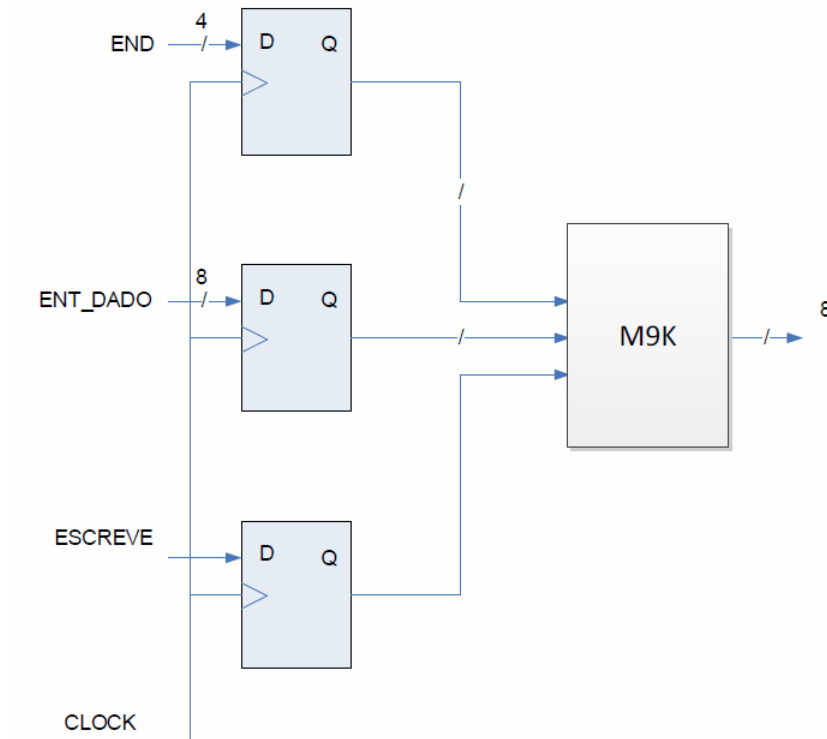
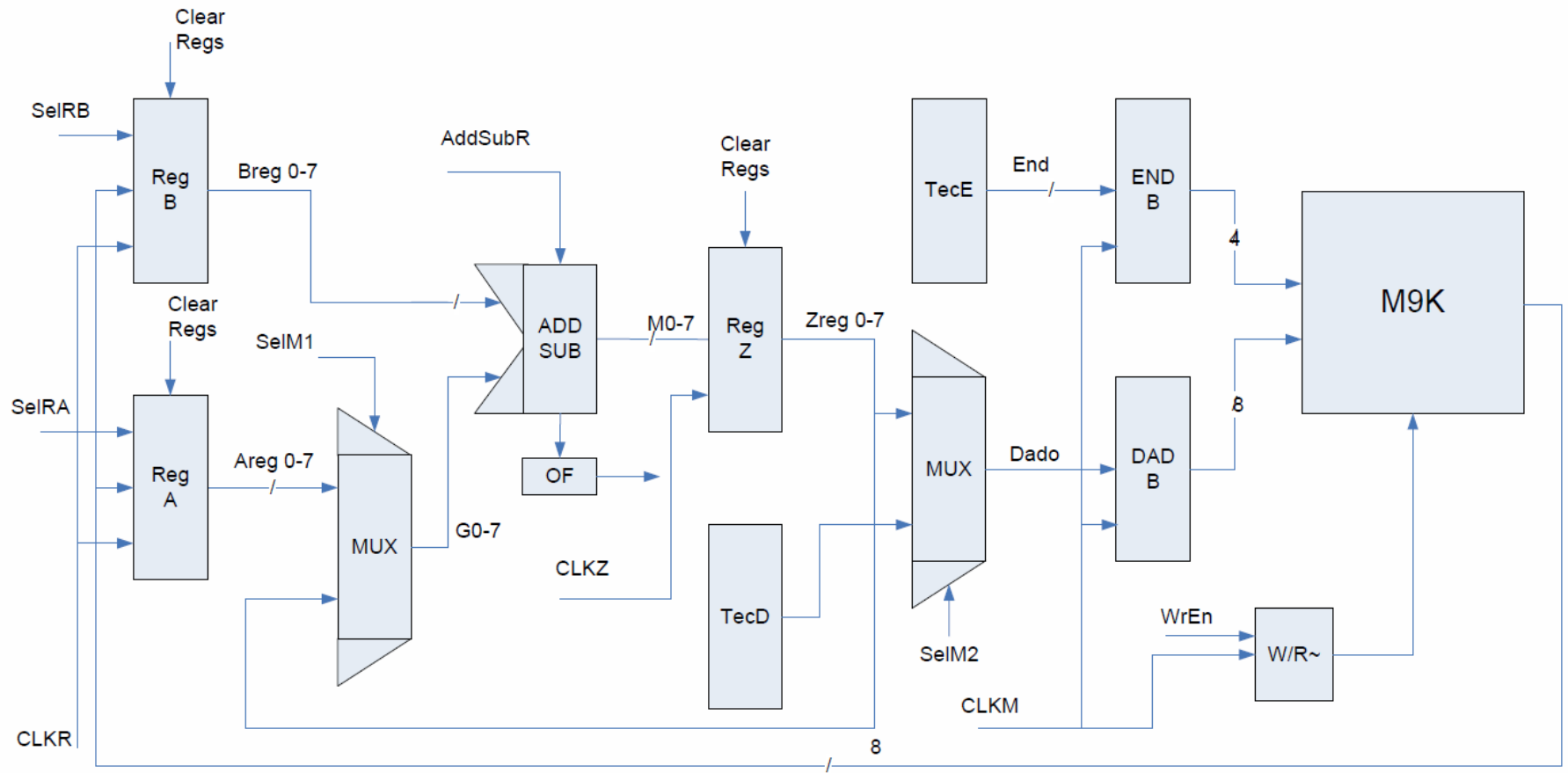


Figura 5: Circuito de memória RAM a ser implementado.

## Parte II

Crie um novo projeto. Anexe o bloco da Figura 5 conforme a Figura 6, usando o mesmo esquema de chaves e saídas 7-segmentos acima para a atribuição de dados. O diagrama da figura 6 deve ser implementado por completo para este experimento, pois o dado a ser armazenado na memória RAM pode vir tanto do REG Z quanto da entrada de dados. Use os componentes apresentados no texto associado. Mostre que consegue realizar a soma/subtração entre RegA e RegB, armazenar e ler diversas posições da memória. Fique à vontade para determinar a função dos leds a fim de facilitar o debug. Envie este código ao Ensino Aberto.



Esquema de unidade aritmética com registros e memória