

Neste manual, podem ser usados nomes de marcas registradas. Caso alguma marca não receba o símbolo de marca registrada, o que se dará somente por enganos, e mesmo em relação às marcas mencionadas seguidas dos símbolos de marca registrada, declaramos que estamos usando os nomes somente de forma editorial e sem nenhuma intenção de violação.

SUMÁRIO

Capítulo 1 - Conteúdo

1.1 Conteúdo do kit	7
1.2 Instalando o kit	7
1.3 Suporte	7

Capítulo 2 - Placa Mercurio® IV

2.1 Descrição geral das PLACAS DO KIT	8
2.2 Sistemas da placa Mercurio® IV	9
2.3 Utilizando o kit Mercurio® IV	11
2.3.1 Configurando o FPGA	11
2.3.2 Utilizando as chaves e os botões	15
2.3.3 Utilizando o display de 7 segmentos	17
2.3.4 Utilizando a matriz de LEDs	19
2.3.5 Utilizando o LED RGB de alto brilho	21
2.3.6 Utilizando o sensor de temperatura	22
2.3.7 Utilizando o display LCD	23
2.3.8 Utilizando a porta VGA	25
2.3.9 Utilizando a porta RS-232	27
2.3.10 Utilizando a porta USB	28
2.3.11 Utilizando o conversor AD	30
2.3.12 Utilizando o conversor DA	32
2.3.13 Utilizando o cartão MicroSD	34
2.3.14 Utilizando a memória Flash	36
2.3.15 Utilizando a memória SDRAM	37
2.3.16 Utilizando a interface Ethernet	40
2.3.17 Utilizando as interfaces PMOD®	42
2.3.18 Utilizando as interfaces GPIO	44
2.3.19 Utilizando a placa expansora com 16 chaves	48

Capítulo 3

3.1 - Histórico de revisões.....	50
3.2 - Copyright.....	50

ÍNDICE DE FIGURAS

Figura 1 - Componentes da placa Mercurio® IV	8
Figura 2 - Diagrama de blocos da placa Mercurio® IV	9
Figura 3 - Chave Liga/Desliga na Posição OFF	12
Figura 4 - Chave Prog na Posição FPGA.....	12
Figura 5 - Chave Prog na Posição Flash	13
Figura 6 - Iniciando o conversor de arquivos do Quartus® II	13
Figura 7 - Tela de conversão de arquivos.....	14
Figura 8 - Tela de conversão de arquivos.....	14
Figura 9 - Ativando a geração automática do arquivo .pof	15
Figura 10 - Chaves do tipo Slide Switch e relação entre posição e nível lógico.....	15
Figura 11 - Conexão entre os Push Buttons e o FPGA.....	16
Figura 12 - Conexão entre os displays e o FPGA	18
Figura 13 - Disposição das linhas e colunas da matriz de LEDs	20
Figura 14 - Conexão entre o LED RGB e o FPGA.....	22
Figura 15 - Conversão DA para a saída VGA.....	25
Figura 16 - Componente de conversão para RS-232	27
Figura 17 - Formato comumente utilizado em transmissões seriais assíncronas	27
Figura 18 - FTDI entre o conector USB e o barramento de entrada do FPGA na placa Mercurio® IV	28
Figura 19 - Sinais para leitura de dado da FIFO do FTDI FT245RQ	29
Figura 20 - Sinais para escrita de dado na FIFO do FTDI FT245RQ.....	29
Figura 21 - Forma de onda do conversor AD no modo dual	30
Figura 22 - Forma de onda do conversor DA.....	34
Figura 23 - Forma de onda da leitura do registrador de status da flash.....	37
Figura 24 - Forma de onda do comando Ativar.....	39
Figura 25 - Jumpers de seleção de tensão.....	42
Figura 26 - Interfaces PMOD® e funções dos pino	43
Figura 27 - Conectores GPIO 0 e GPIO 1 e a descrição de seus pinos	45

ÍNDICE DE TABELAS

Tabela 1 - Pinagem do FPGA relacionada a chaves e botões	17
Tabela 2 - Pinagem do FPGA relacionada aos displays	18
Tabela 3 - Pinagem do FPGA relacionada à matriz de LEDs	21
Tabela 4 - Pinagem do FPGA relacionada ao LED RGB.....	22
Tabela 5 - Pinagem do FPGA relacionada ao sensor de temperatura.....	23
Tabela 6 - Mapeamento do display na memória DDRAM.....	24
Tabela 7 - Pinagem do FPGA relacionada ao display LCD 16x2	24
Tabela 8 - Temporização horizontal (linha)	26
Tabela 9 - Temporização vertical (frame)	26
Tabela 10 - Pinagem do FPGA relacionada à porta VGA.....	26
Tabela 11 - Pinagem do FPGA relacionada à porta serial	28
Tabela 12 - Pinagem do FPGA relacionada à porta USB.....	30
Tabela 13 - Pinagem do FPGA relacionada ao conversor DA	31
Tabela 14 - Comandos de escrita do DAC	33
Tabela 15 - Configuração do DAC	33
Tabela 16 - Pinagem do FPGA relacionada ao conversor DA	34
Tabela 17 - Pinagem do FPGA relacionada ao cartão MicroSD	35
Tabela 18 - Comandos SPI da memória Flash.....	36
Tabela 19 - Pinagem do FPGA relacionada à memória Flash.....	37
Tabela 20 - Algumas operações sobre a memória SDRAM.....	38
Tabela 21 - Pinagem do FPGA relacionada à memória SDRAM	39
Tabela 22 - Pinagem do FPGA relacionada à interface Ethernet	41
Tabela 23 - Pinagem do FPGA relacionada às interfaces PMOD®	43
Tabela 24 - Pinagem do FPGA relacionada às interfaces GPIO.....	45
Tabela 25 - Pinagem da placa expansora	49

CAPÍTULO 1 – CONTEÚDO

O kit Mercurio® IV possui todo o necessário para a utilização do mesmo. O usuário deve apenas fornecer alguma fonte de alimentação de energia, através de uma porta USB ou de uma fonte de alimentação (não inclusa no kit). O usuário pode ainda utilizar um computador (não incluso no kit) para alimentar a placa através da porta USB e para configurar o FPGA da placa através da USB-Blaster™. A alimentação via USB e a configuração, via USB-Blaster™, são feitas pelo mesmo conector, de forma transparente ao usuário.

1.1 CONTEÚDO DO KIT

O kit de desenvolvimento Mercurio® IV consiste dos seguintes itens:

- Placa Mercurio® IV;
- Placa Expansora com 16 chaves;
- Cabo USB de programação e alimentação;
- Exemplos e referências;
- Protetor de acrílico para a placa.

1.2 INSTALANDO O KIT

- Conectar uma fonte de alimentação à placa Mercurio® IV. A placa pode ser alimentada de duas formas diferentes:

- Conectar um cabo USB no conector USB-Blaster™ (a esquerda da placa, do lado do display LCD) a um computador ou tomada USB;
- Conectar uma fonte de 5V no conector de alimentação externa (extremidade superior esquerda da placa, acima da matriz de LEDs).

- Ao ligar a placa Mercurio® IV pela primeira vez, ela iniciará em modo de demonstração, no qual algumas interfaces existentes na placa serão ligadas. O arquivo de demonstração está disponível para download no site da Macnica DHW ou pode ser solicitado por e-mail.

- A configuração do FPGA pode ser feita somente através do conector USB-Blaster™, quando este estiver conectado à um computador com o software Altera Quartus®.

1.3 SUPORTE

Qualquer problema durante a utilização do kit de desenvolvimento Mercurio® IV, entrar em contato com: Macnica DHW Ltda.



Rua Patrício Farias, 131, Ed. Multicenter, Loja 01, CEP: 88034-132
Itacorubi, Florianópolis, SC, Brasil
Telefone: +55 (48) 3225.5052
Email: suporte@macnicadhw.com.br

CAPÍTULO 2 - PLACA MERCURIO® IV

2.1 DESCRIÇÃO GERAL DAS PLACAS DO KIT

A placa Mercurio® IV é mostrada na Figura 1, nessa figura estão indicados os diversos periféricos que poderão ser utilizados.

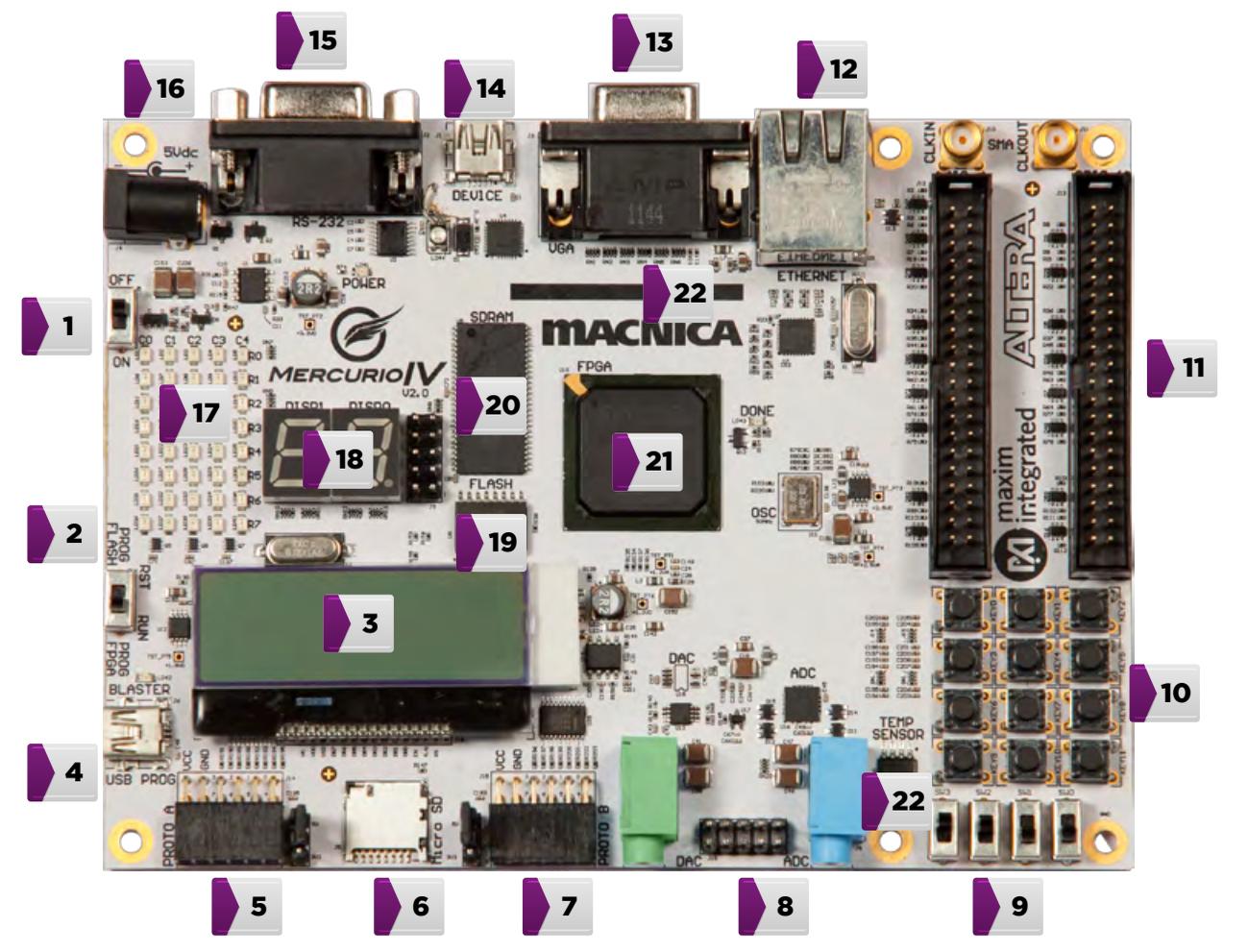


FIGURA 1 - COMPONENTES DA PLACA MERCURIO® IV

A seguir estão listados os itens que compõem a placa Mercurio® IV:

- | | |
|----------------------------------|--|
| 1 - Liga / Desliga; | 12 - Ethernet; |
| 2 - Modo de configuração; | 13 - VGA; |
| 3 - Display LCD; | 14 - USB; |
| 4 - USB-Blaster™; | 15 - Serial; |
| 5 e 7 - Interfaces PMOD®; | 16 - LED RGB de alto brilho; |
| 6 - Cartão micro SD; | 17 - Matriz de LEDs; |
| 8 - ADs e DA;s; | 18 - Displays de 7 segmentos; |
| 9 - Switches; | 19 e 20 - Memórias Flash e RAM; |
| 10 - Teclado numérico; | 21 - FPGA Cyclone® IV; |
| 11 - GPIO; | 22 - Sensor de temperatura |

A placa expansora contida no kit possui 16 chaves e é compatível com as interfaces GPIO da placa Mercurio® IV.

2.2 SISTEMAS DA PLACA MERCURIO® IV

Para facilitar no desenvolvimento de diferentes sistemas, todos os subsistemas são interligados pelo FPGA Cyclone® IV, assim, o desenvolvedor pode configurar o FPGA para apenas controlar os periféricos necessários, sem se preocupar com os demais itens.



FIGURA 2 - DIAGRAMA DE BLOCOS DA PLACA MERCURIO® IV

FPGA CYCLONE® IV

O FPGA que equipa a placa Mercurio® IV é uma Cyclone® IV EP4CE30F23, a qual possui 30 mil elementos lógicos.

O clock de entrada do FPGA é de 50MHz.

SWITCHES

O kit dispõe de 4 chaves do tipo slide switch, que quando estão na posição para baixo, geram sinal de nível lógico baixo nas entradas do FPGA.

TECLADO NUMÉRICO

Na placa Mercurio® IV existe uma organização de 12 botões do tipo push button a fim de simular um teclado telefônico. Os botões não pressionados geram sinal de nível lógico baixo na entrada do FPGA, e quando apertados geram um pulso de nível lógico alto.

DISPLAYS DE 7 SEGMENTOS

A placa Mercurio® IV possui dois displays de sete segmentos. Cada display é controlado por um barramento próprio de dados, sendo assim, displays

independentes entre si.

MATRIZ DE LEDS

O kit de desenvolvimento Mercurio® IV conta com uma matriz de LEDs disposta em 8 linhas e 5 colunas. Devido a essa disposição dos LEDs, o acesso a eles deve ser multiplexado, com cada coluna (ou linha) sendo acessada por vez.

LED RGB DE ALTO BRILHO

O LED RGB de alto brilho presente na placa Mercurio® IV é capaz de emitir as três frequências básicas (RGB) e a combinação entre elas.

DISPLAY LCD

Neste kit também existe um display LCD 16 x 2 caracteres, e com possibilidade de ligar iluminação de fundo.

SENSOR DE TEMPERATURA

Foi instalado um sensor de temperatura na placa Mercurio® IV. O FPGA pode se comunicar com o sensor através de dois pinos, pelo protocolo I2C.

VGA

Com a saída VGA é possível conectar a placa Mercurio® IV em um monitor de vídeo e exibir imagens, que podem ser processadas pelo FPGA.

Cada componente, de 4 bits, é convertida em sinal analógico através de uma rede de resistores e o conector utilizado é o padrão D-SUB.

CARTÃO MICRO SD

A placa Mercurio® IV é capaz de ler e escrever cartões de memória do formato micro SD. O acesso ao cartão pode ser SPI ou no modo Cartão SD de até 4-Bits.

CONVERSOR AD (ANALÓGICO-DIGITAL)

Uma das características que mais destaca o kit Mercurio® IV em relação ao seus similares é a presença dos conversores de sinal. O conversor analógico-digital disponível neste kit possui 2 entradas diferenciais, ou 4 entradas single ended, 2 canais de conversão paralelos de 12 bits de resolução, que operam em até 1,25 Msps. Os resultados das conversões são apresentados pelo barramento serial SPI.

CONVERSOR DA (DIGITAL-ANALÓGICO)

Para gerar sinais analógicos, como por exemplo um sinal de áudio, o kit Mercurio® IV vem equipado com dois canais de conversão digital-analógico. Assim como os conversores AD, esses conversores DA tem resolução de 12 bits e se comunicam pelo barramento SPI, serialmente.

MEMÓRIA SDRAM

A memória SDRAM instalada na placa Mercurio® IV possui 512 Mbits. Essa é a memória de acesso rápido da placa Mercurio® IV.

MEMÓRIA FLASH

A memória Flash utilizada para EPCS e para armazenamento de dados, tem tamanho de 64 Mbits e possui acesso serial.

ETHERNET

A porta Ethernet pode ser utilizada em conjunto com o softcore NIOS® para se construir um servidor web, por exemplo.

RS-232

A porta serial fornecida com o kit Mercurio® IV é usada para realizar comunicações com outros dispositivos serial, como a porta serial de um PC.

USB-BLASTER™

A porta USB-Blaster™ pode ser utilizada para alimentar a placa Mercurio® IV e, além disso, é por ela que é feita a programação e comunicação com o FPGA da placa. É importante ressaltar que a única forma de configurar o FPGA é pela porta USB-Blaster™.

USB

Outro conector USB existe na placa Mercurio® IV. Chamado de USB Device, ele pode ser utilizado para estabelecer uma conexão USB entre o FPGA da placa com periféricos hosts USB externos.

INTERFACES PMOD®

São fornecidas duas interfaces para periféricos compatíveis com interface PMOD®. A comunicação entre a placa Mercurio® IV e esses periféricos pode ser de diferentes maneiras, por exemplo, o sensor de proximidade tem comunicação I2C, já o thermocouple para digital se comunica por SPI. As placas PMOD® devem ser adquiridas separadamente.

INTERFACES GPIOS

Além das interfaces para periféricos PMOD®, existem dois barramentos de pinos de propósito geral, cada um com 32 pinos de I/O.

2.3 UTILIZANDO O KIT MERCURIO® IV

Nesta sessão será mostrado como o usuário poderá utilizar o kit Mercurio® IV, desde a configuração do FPGA até detalhamentos do funcionamento de cada periférico disponível na placa.

2.3.1 CONFIGURANDO O FPGA

A configuração do FPGA Cyclone® IV do kit Mercurio® IV pode ser feita por dois modos, o modo JTAG e o modo AS (não é suportado o modo Active Serial x4).

- O modo JTAG faz a configuração diretamente no FPGA, que irá manter essa configuração enquanto a placa estiver sendo alimentada. Quando a placa for desligada, essa configuração será perdida.
- Pelo modo AS, a configuração do FPGA é enviada para a memória EPCS64,

assim, essa configuração será mantida numa memória não volátil e não será perdida quando a placa for desligada. Quando a placa é ligada, a configuração do FPGA situada na memória EPCS é imediatamente transferida para o FPGA, que inicia sua operação ao terminar a transferência.

Na placa Mercurio® IV a memória EPCS64 é uma memória Flash, que também pode ser utilizada para armazenamento de dados do usuário.

Para iniciar a configuração da placa Mercurio® IV, conecte uma extremidade do cabo USB no conector USB-Blaster™ da placa, e a outra extremidade num computador equipado com o software Altera Quartus® II. Com o cabo conectado, e o computador ligado, a placa automaticamente será alimentada, então basta acionar a chave Liga/Desliga, na posição ON, para ligar a placa. Quando a placa for ligada, o FPGA será configurado de acordo com os dados da memória EPCS64, que por padrão é um projeto que ativa diversos sistemas da placa (LEDs, displays, botões, etc.).

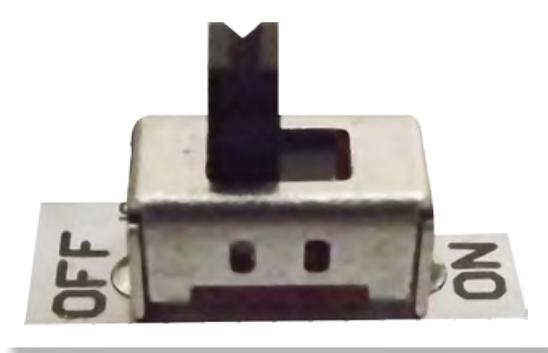


FIGURA 3 - CHAVE LIGA/DESLIGA NA POSIÇÃO OFF

CONFIGURANDO PELO MODO JTAG

Com a placa ligada e conectada ao computador, se o usuário deseja realizar a configuração pelo JTAG, a chave Prog deve ser colocada na posição FPGA (esse também é o modo de execução do FPGA). Então basta utilizar o módulo de programação do Quartus® II (Quartus® II Programmer) para transferir a configuração do projeto, armazenada no arquivo de extensão .sof, para o FPGA.

O arquivo .sof é gerado automaticamente, pelo Quartus® II.



FIGURA 4 - CHAVE PROG NA POSIÇÃO FPGA

CONFIGURANDO PELO MODO AS

Já para o modo Active Serial de programação, a chave Prog deve ser colocada na posição FLASH, o que significa que o arquivo de configuração será transferido para a memória EPCS64.

Com a placa no modo FLASH, o usuário deve utilizar o módulo de programação do Quartus® II para transferir o arquivo de configuração, dessa vez um arquivo com extensão .pof.



FIGURA 5 - CHAVE PROG NA POSIÇÃO FLASH

O arquivo .pof não é gerado por padrão no Quartus® II. Existem duas possibilidades para se gerar esse arquivo, a primeira é o modo manual, na qual o usuário converte o arquivo .sof para um arquivo .pof que ele escolher, a segunda possibilidade é configurar o Quartus® II para automaticamente gerar o arquivo .pof durante a compilação do projeto, assim, após a compilação, ambos os arquivos .sof e .pof estarão disponíveis para a configuração do FPGA.

- O modo manual é feito através de um aplicativo do Quartus® II de conversão de arquivos. Para iniciá-lo, entre no menu 'File' e selecione a entrada 'Convert Programming Files...'

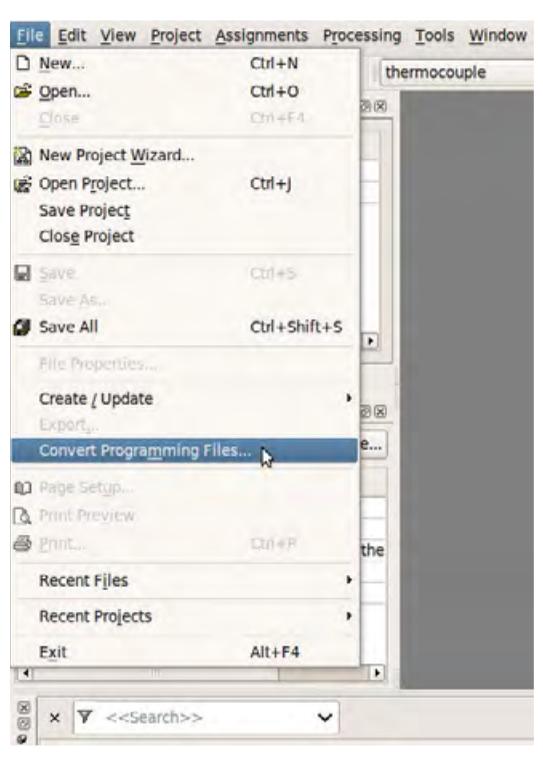


FIGURA 6 - INICIANDO O CONVERSOR DE ARQUIVOS DO QUARTUS® II

A seguir, na janela do conversor de arquivos, o usuário deve definir o dispositivo de configuração para EPCS64 no modo Active Serial. Então define o nome do arquivo .pof de saída e adiciona o arquivo .sof que deve ser convertido através do

botão 'Add File', que fica ativo quando uma entrada da lista 'Input files to convert' está selecionada.

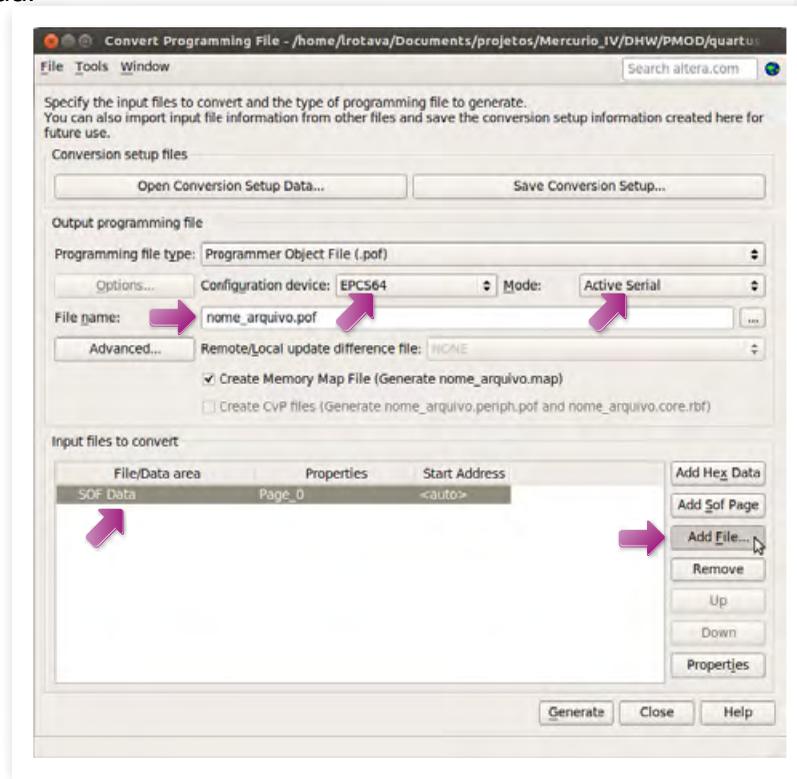


FIGURA 7 - TELA DE CONVERSÃO DE ARQUIVOS

Após selecionar um arquivo .sof, esse arquivo será listado abaixo da linha 'SOF Data' selecionada na lista de entrada.

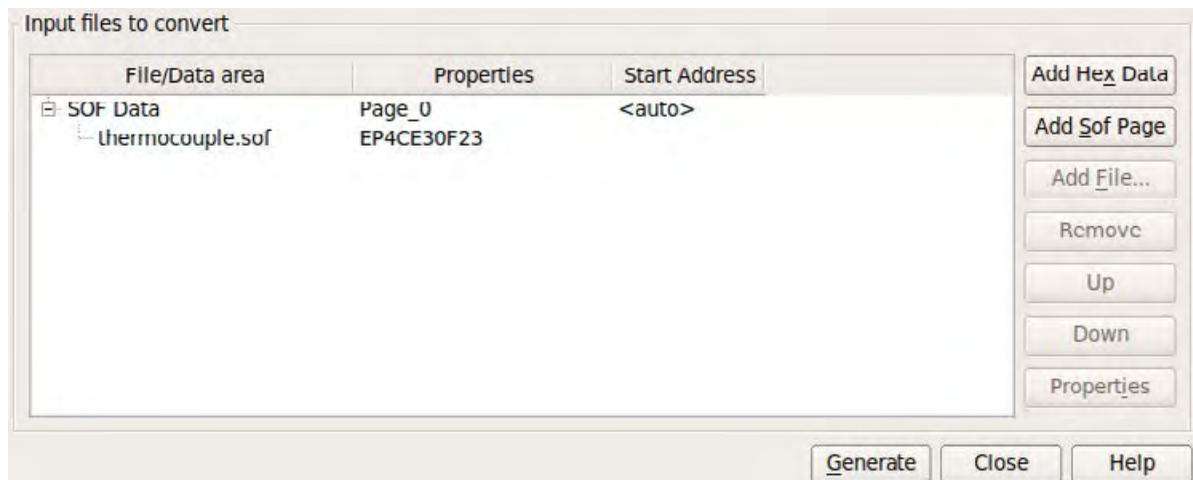


FIGURA 8 - TELA DE CONVERSÃO DE ARQUIVOS

Com o arquivo .sof selecionado, basta clicar no botão 'Generate' e o arquivo .pof será gerado.

- Para o Quartus® criar o arquivo .pof automaticamente, entre no menu 'Assignments' e selecione a entrada 'Device...'. Na janela que se abriu clique no botão 'Device and Pin Options...', o que abrirá uma nova janela. Então, entre na categoria 'Configuration', marque a opção 'Use configuration device:' e selecione o

dispositivo EPCS64.

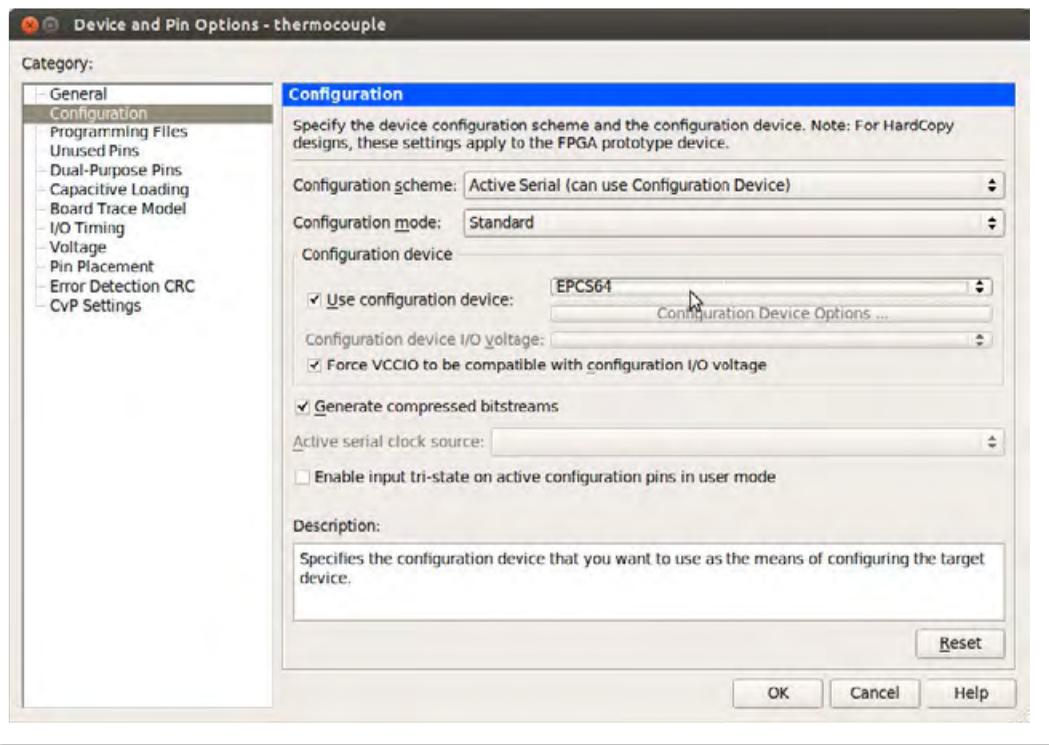


FIGURA 9 - ATIVANDO A GERAÇÃO AUTOMÁTICA DO ARQUIVO .POF

Agora, durante a compilação, o Quartus® II gerará tanto o arquivo .sof quanto o arquivo .pof.

2.3.2 UTILIZANDO AS CHAVES E OS BOTÕES

A placa Mercurio® IV possui quatro chaves do tipo Slide Switch que são conectadas diretamente ao FPGA. Se a chave estiver posicionada na parte de baixo, o nível lógico na porta do FPGA será '0', se a chave estiver posicionada para cima o nível lógico produzido será '1'. Elas estão ligadas a filtros RC, porém, mesmo assim, pode ser interessante implementar um processo de debounce do sinal, a fim de evitar glitches na troca de posição das chaves.



FIGURA 10 - CHAVES DO TIPO SLIDE SWITCH E RELAÇÃO ENTRE POSIÇÃO E NÍVEL LÓGICO

Além das chaves, a placa Mercurio® IV possui doze botões do tipo Push button, que estão arranjados de forma a simular um teclado telefônico. Esses botões, quando não pressionados, mantêm na entrada do FPGA sinal '0', portanto, quando algum botão for pressionado, será gerado um pulso em alto, '1', na respectiva

entrada do FPGA.

Existe um filtro RC acoplado à esses botões para diminuir o ruído nas linhas, porém seu efeito é muito pequeno considerando a transição de estados, portanto é altamente recomendado que seja implementado no FPGA rotinas de debounce para esses botões.

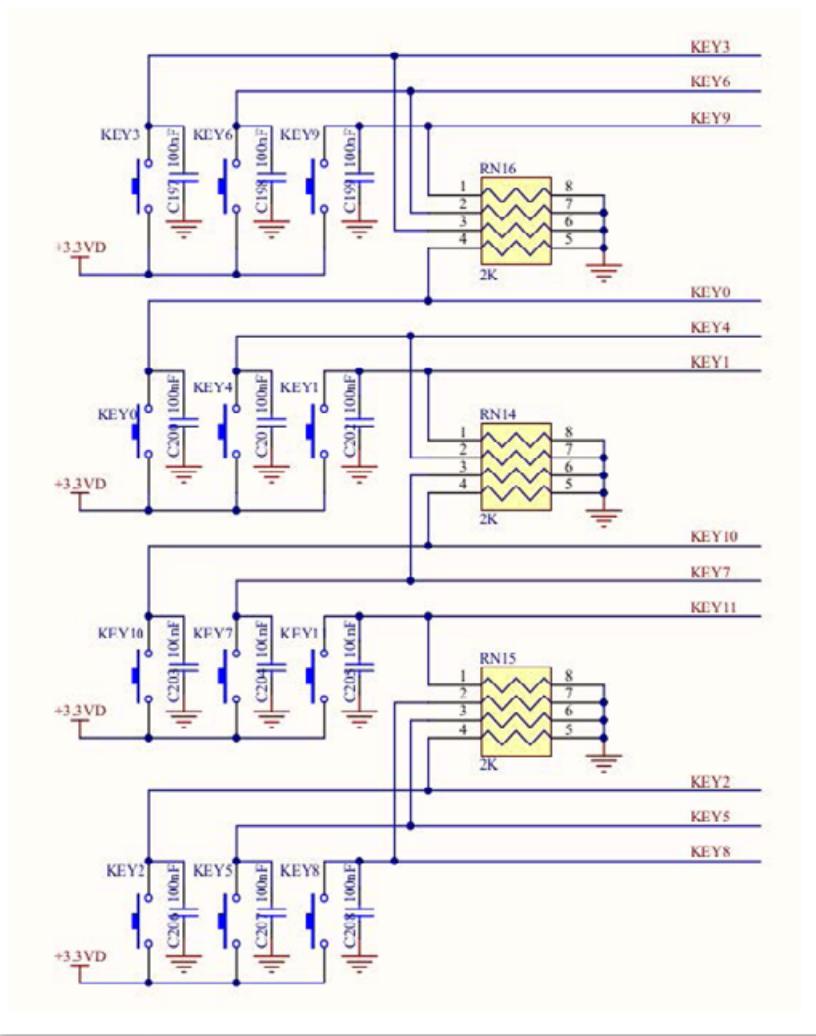


FIGURA 11 - CONEXÃO ENTRE OS PUSH BUTTONS E O FPGA

Como indicado nas figuras, as chaves serão chamadas de SW[3..0] e os botões de KEY[11..0]. Dito isto, o mapeamento dos pinos do FPGA com relação às chaves e aos botões é dado pela Tabela 1.

TABELA 1 - PINAGEM DO FPGA RELACIONADA A CHAVES E BOTÕES

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
CHAVES TIPO SLIDE SWITCH		
SW[0]	V21	Chave tipo Slide Switch 0
SW[1]	W22	Chave tipo Slide Switch 1
SW[2]	W21	Chave tipo Slide Switch 2
SW[3]	Y22	Chave tipo Slide Switch 3
BOTÕES TIPO PUSH BUTTON		
KEY[0]	V22	Botão tipo Push Button 0
KEY[1]	U20	Botão tipo Push Button 1
KEY[2]	U22	Botão tipo Push Button 2
KEY[3]	U16	Botão tipo Push Button 3
KEY[4]	W20	Botão tipo Push Button 4
KEY[5]	U21	Botão tipo Push Button 5
KEY[6]	V15	Botão tipo Push Button 6
KEY[7]	W17	Botão tipo Push Button 7
KEY[8]	W19	Botão tipo Push Button 8
KEY[9]	W15	Botão tipo Push Button 9
KEY[10]	U17	Botão tipo Push Button 10
KEY[11]	Y17	Botão tipo Push Button 11

REFERÊNCIA 1 - CHAVES E BOTÕES

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
SW0 - SW3	SPDT Subminiature Toggle Switch, Right Angle Mounting, Vertical Actuation	-	A06-5.0
KEY0 - KEY11	SPST Tactil Switch	-	OS102011MS2QN1

2.3.3 UTILIZANDO O DISPLAY DE 7 SEGMENTOS

Na placa Mercurio® IV existem dois displays de 7 segmentos arranjados um ao lado do outro, para ser possível exibir um número com dois algarismos. O display DISPO fica a direita e o DISP1 a esquerda. Cada display possui um barramento de oito sinais ligado no FPGA, sete sinais para os segmentos e um sinal para o ponto decimal. Os displays são do tipo Anodo Comum, portanto, nível '1' no pino de entrada acende o segmento do display e nível '0' apaga o segmento.

A relação entre segmentos e sinais do FPGA é ilustrada pela Figura 12.

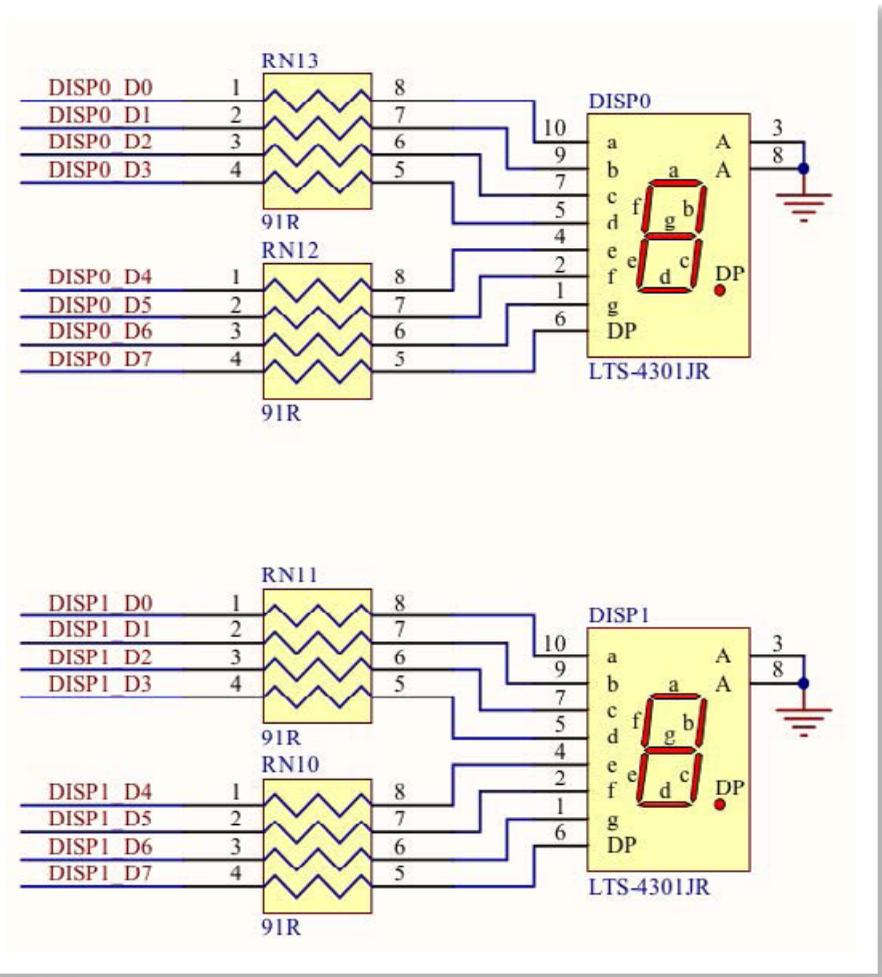


FIGURA 12 - CONEXÃO ENTRE OS DISPLAYS E O FPGA

TABELA 2 - PINAGEM DO FPGA RELACIONADA AOS DISPLAYS

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
DISPLAY 0		
DISPO_D[0]	V2	Segmento horizontal superior do display 0
DISPO_D[1]	V1	Segmento vertical superior direito do display 0
DISPO_D[2]	U2	Segmento vertical inferior direito do display 0
DISPO_D[3]	U1	Segmento horizontal inferior do display 0
DISPO_D[4]	Y2	Segmento vertical inferior esquerdo do display 0
DISPO_D[5]	Y1	Segmento vertical superior esquerdo do display 0
DISPO_D[6]	W2	Segmento horizontal do meio do display 0
DISPO_D[7]	W1	Ponto decimal do display 0

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
DISPLAY 1		
DISP1_D[0]	R5	Segmento horizontal superior do display 1
DISP1_D[1]	T5	Segmento vertical superior direito do display 1
DISP1_D[2]	T3	Segmento vertical inferior direito do display 1
DISP1_D[3]	T4	Segmento horizontal inferior do display 1
DISP1_D[4]	M6	Segmento vertical inferior esquerdo do display 1
DISP1_D[5]	N7	Segmento vertical superior esquerdo do display 1
DISP1_D[6]	N6	Segmento horizontal do meio do display 1
DISP1_D[7]	P6	Ponto decimal do display 1

REFERÊNCIA 2 - DISPLAY DE 7 SEGMENTOS

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
DISP0 e DISP1	10.16 mm Black Surface HER 7- Segment Display: CA, RH DP	-	LTS-4301JR

2.3.4 UTILIZANDO A MATRIZ DE LEDs

A matriz de LEDs da placa Mercurio® IV é composta por oito linhas e cinco colunas de LEDs, conforme a Figura 13. Cada linha e cada coluna está conectada a um pino do FPGA, portanto, para acender apenas o LED superior esquerdo, por exemplo, é necessário colocar a linha superior e a coluna esquerda em nível baixo, os demais sinais devem permanecer em nível alto (lógica invertida). Ou seja, o acesso à matriz deve ser multiplexado, ou por linhas ou por colunas.

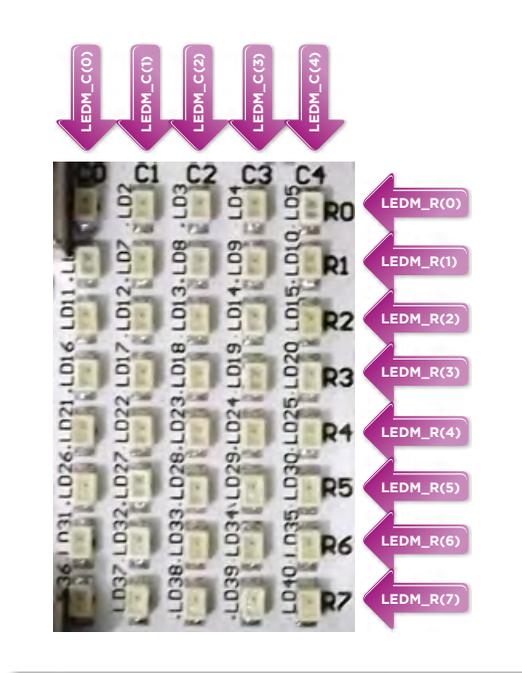


FIGURA 13 - DISPOSIÇÃO DAS LINHAS E COLUNAS DA MATRIZ DE LEDS

Adotando a multiplexação por colunas como exemplo (são menos colunas que linhas, portanto é recomendada a multiplexação por colunas nesse caso), num primeiro instante coloca-se a primeira coluna em nível baixo e as demais em nível alto, assim somente os LEDs da primeira coluna podem ser acesos. Então coloca-se as linhas desejadas em nível baixo, para acender apenas os LEDs da primeira coluna e das linhas desejadas.

Num segundo instante, a primeira coluna é colocada em nível alto (apagando momentaneamente os LEDs), e a segunda coluna é colocada em nível baixo. Assim, ativando as linhas desejadas é possível acender os LEDs da segunda coluna.

Repete-se o processo até a última coluna, e então reinicia-o pela primeira coluna, enquanto a matriz de LEDs estiver sendo utilizada. É importante que o tempo de acendimento de cada coluna seja suficiente para que os LEDs sejam visivelmente acesos, porém esse tempo não deve ser muito longo, para que o observador não consiga perceber que a coluna foi apagada (até ser acesa novamente).

TABELA 3 - PINAGEM DO FPGA RELACIONADA À MATRIZ DE LEDS

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
LEDM_C[0]	J7	Coluna 0 da matriz de LEDs
LEDM_C[1]	J6	Coluna 1 da matriz de LEDs
LEDM_C[2]	K8	Coluna 2 da matriz de LEDs
LEDM_C[3]	J8	Coluna 3 da matriz de LEDs
LEDM_C[4]	L8	Coluna 4 da matriz de LEDs
LEDM_R[0]	F10	Linha 0 da matriz de LEDs
LEDM_R[1]	C8	Linha 1 da matriz de LEDs
LEDM_R[2]	E9	Linha 2 da matriz de LEDs
LEDM_R[3]	G9	Linha 3 da matriz de LEDs
LEDM_R[4]	F9	Linha 4 da matriz de LEDs
LEDM_R[5]	F8	Linha 5 da matriz de LEDs
LEDM_R[6]	G8	Linha 6 da matriz de LEDs
LEDM_R[7]	H11	Linha 7 da matriz de LEDs

REFERÊNCIA 3 - LEDS DA MATRIZ DE LEDS

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
LD1 - LD40	High Performance Chip LED	-	LS M67K-H2L1-1-Z

2.3.5 UTILIZANDO O LED RGB DE ALTO BRILHO

O LED RGB de alto brilho possui três elementos internos, o vermelho, o verde e o azul. Esses três elementos são utilizados para que o LED possa emitir sete cores diferentes:

- Vermelho - elemento R;
- Verde - elemento G;
- Azul - elemento B;
- Amarelo - elementos R e G;
- Ciano - elementos G e B;
- Magenta - elementos R e B;
- e Branco - elementos R, G e B.

Cada elemento possui um pino associado e existe também um pino comum aos três elementos, portanto. Os três elementos são conectados ao FPGA e o pino comum é conectado ao VCC. Assim o FPGA pode ativar ou desativar cada elemento do LED RGB colocando '0' no pino associado, sendo possível gerar as sete cores do LED.

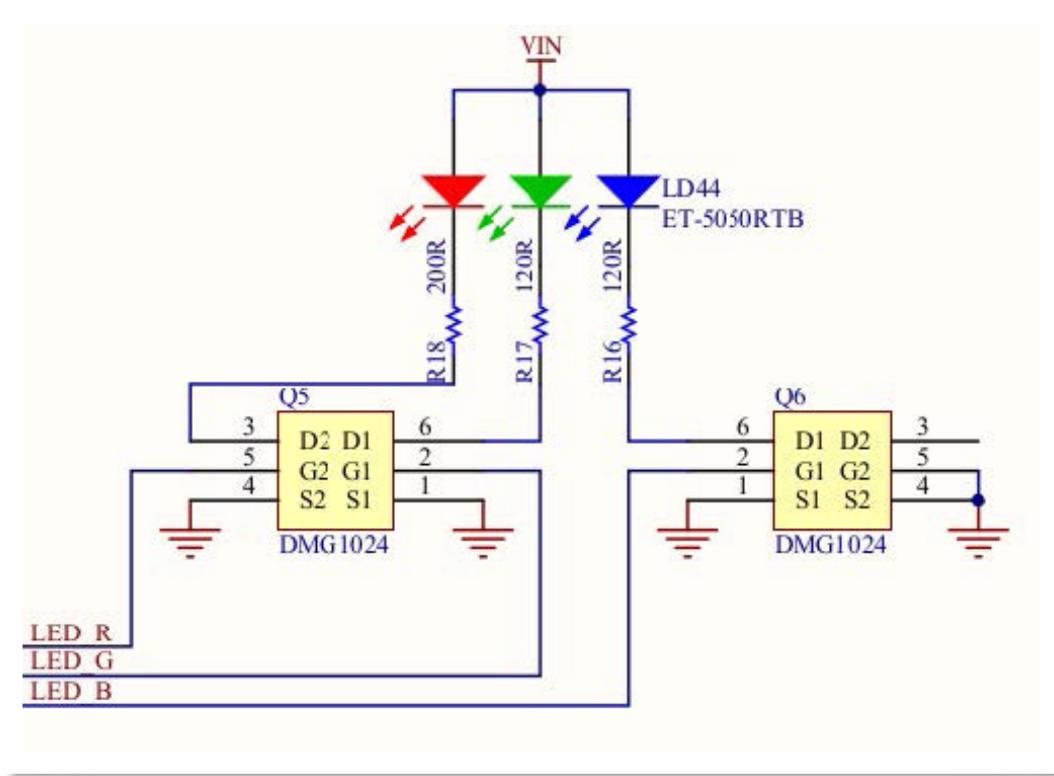


FIGURA 14 - CONEXÃO ENTRE O LED RGB E O FPGA

TABELA 4 - PINAGEM DO FPGA RELACIONADA AO LED RGB

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
LED_R	D6	Elemento vermelho do LED RGB
LED_G	D7	Elemento verde do LED RGB
LED_B	E7	Elemento azul do LED RGB

REFERÊNCIA 4 - LED RGB

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PARTNUMBER
LD44	High Luminous Intensity True Red- Green-Blue LED	-	ET-5050RTB-313

2.3.6 UTILIZANDO O SENSOR DE TEMPERATURA

Está presente na placa Mercurio® IV um sensor de temperatura, que pode ser utilizado para determinar se o equipamento excedeu determinada temperatura. Essa função é utilizada em equipamentos eletrônicos para evitar que qualquer dano ocorra devido ao excesso de temperatura do equipamento.

A comunicação com o sensor é feita por I2C, ou seja, dois pinos do FPGA são utilizados para realizar leituras e configurações do sensor. Além disso existe mais um pino, que pode ser utilizado como termostato, que indica quando a temperatura excedeu um limite superior, a saída OS vai para '1', ou quando excedeu um limite inferior (para menos que o limite), a saída OS vai para '0'. A saída OS também pode ser utilizada como interrupção, para indicar que a temperatura excedeu um

limite superior, nesse modo, a saída OS vai para '0' quando for feita uma leitura, de qualquer registrador, do sensor.

Para fazer a leitura do valor da temperatura, basta que o FPGA faça uma operação de leitura no I2C, informando o endereço do registrador de temperatura (registrador somente leitura). Além do registrador de temperatura existem mais o registrador de configuração e dois registradores que armazenam os valores de temperatura limites superior e inferior.

O acesso à esses registradores não é trivial, por isso recomenda-se a leitura do datasheet do componente antes de utilizá-lo.

TABELA 5 - PINAGEM DO FPGA RELACIONADA AO SENSOR DE TEMPERATURA

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
I2C_SDA	Y21	Sinal de dados da rede I2C com o sensor de temperatura
I2C_SCL	AA21	Sinal de clock da rede I2C com o sensor de temperatura
I2C_OVERTEMP_N	AA20	Saída termostato/interrupção do sensor de temperatura

REFERÊNCIA 5 - SENSOR DE TEMPERATURA

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
U30	256Kbit, 400kHz, 2.5V, I2C Serial EEPROM, 8-Pin SOIC 208mil, Extended Temperature	Maxim Integrated	MAX7500MSA+

2.3.7 UTILIZANDO O DISPLAY LCD

Na placa Mercurio® IV está instalado um display LCD 16x2 caracteres. Esse display possui um mapa de caracteres, que pode ser utilizado para escrever no display. Além do mapa de caracteres, existe uma memória DDRAM utilizada para o mapeamento do display diretamente, ou seja, existem 32 endereços da DDRAM que seus valores são exibidos no display, segundo o mapa de caracteres.

TABELA 6 - MAPEAMENTO DO DISPLAY NA MEMÓRIA DDRAM

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Endereço da DDRAM	Posição no display																
	Linha 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	Linha 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

Por exemplo, no mapa de caracteres o valor, em binário, 00110000 corresponde ao caractere 0, então, se o dado 00110000 for escrito no endereço 0 da memória DDRAM, o caractere 0 será desenhado na primeira posição, superior esquerda, do display.

A configuração do display e a escrita dos caracteres é feita pelo barramento LCD_DATA[7..0] e pelos sinais LCD_EN, LCD_RS e LCD_RW. LCD_RW em nível alto indica que será feita uma leitura e em nível baixo uma escrita, LCD_RS em nível alto indica que o acesso será feito na memória do display e em nível baixo indica que o acesso será feito num registrador de configuração do display. O sinal LCD_EN é colocado em nível alto quando se deseja realizar uma operação.

Resta então o sinal LCD_BACKLIGHT, que é utilizado para ligar ou desligar o backlight do display.

TABELA 7 - PINAGEM DO FPGA RELACIONADA AO DISPLAY LCD 16X2

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
LCD_BACKLIGHT	V10	Controlador do backlight
LCD_EN	V9	Operation Enable, inicia uma operação com o LCD
LCD_RS	U9	Register Select, seleciona se o acesso é em registrador ('0') ou em memória ('1')
LCD_RW	U8	Read Write, indica se é uma leitura ('1') ou escrita ('0')
LCD_D[0]	V8	Bit 0 do barramento de comando/dado - LSB
LCD_D[1]	V7	Bit 1 do barramento de comando/dado
LCD_D[2]	V6	Bit 2 do barramento de comando/dado
LCD_D[3]	V5	Bit 3 do barramento de comando/dado
LCD_D[4]	V4	Bit 4 do barramento de comando/dado
LCD_D[5]	Y4	Bit 5 do barramento de comando/dado
LCD_D[6]	V3	Bit 6 do barramento de comando/dado
LCD_D[7]	Y3	Bit 7 do barramento de comando/dado - MSB

REFERÊNCIA 6 - DISPLAY LCD

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
J18	2x16 characters, 3Vdd, white LED backlight 3V	NewHaven	NHD-CO216CU-FSW-GBW-3V3

2.3.8 UTILIZANDO A PORTA VGA

A porta VGA é utilizada para enviar sinal de vídeo para um monitor compatível. O sinal enviado para o monitor possui uma parte analógica, as componentes de cor, e uma parte digital, os sinais de sincronização.

Os sinais digitais de sincronização são o sincronismo horizontal e o sincronismo vertical. Nível baixo no sincronismo horizontal indica que uma nova linha será iniciada, e nível baixo no sincronismo vertical indica que um novo frame será iniciado. Além desses dois sinais existem três componentes de cor, o vermelho, o verde e o azul, que são sinais analógicos.

Como o FPGA é puramente digital, as cores devem ser convertidas de digital para analógico, e isso é feito na placa Mercurio® IV com um conversor DA resistivo. O FPGA gera um sinal digital de 4 bits para cada componente de cor, e através de uma associação resistiva, esse valor de 4 bits é convertido em um sinal analógico.

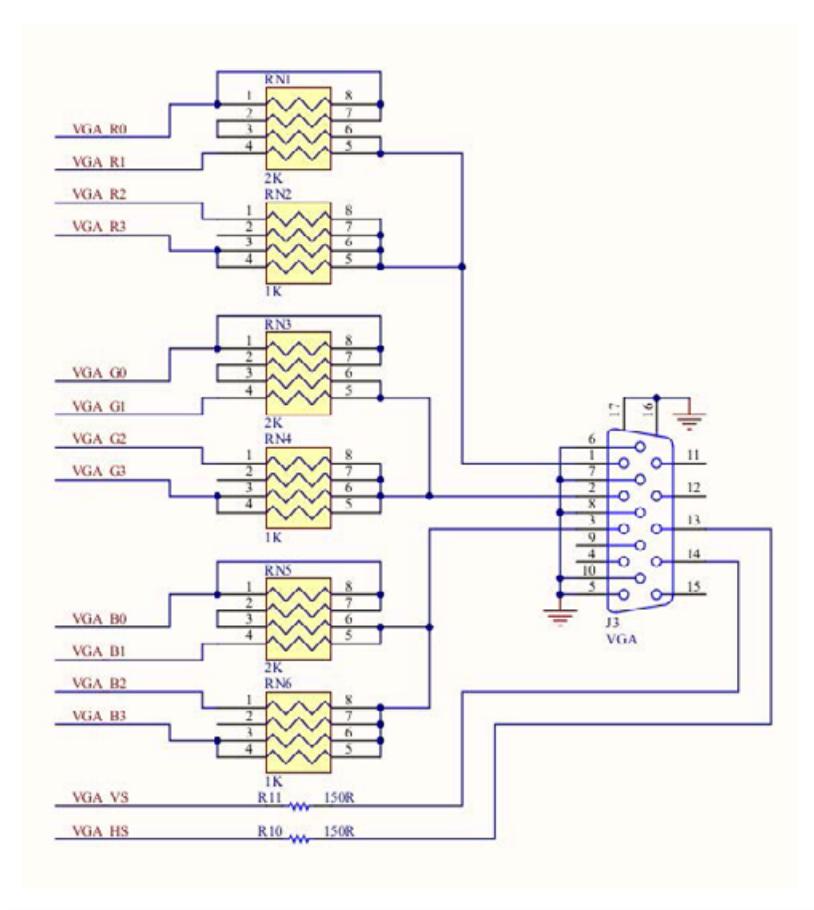


FIGURA 15 - CONVERSÃO DA PARA A SAÍDA VGA

Todos os sinais devem ser sincronizados por um clock de pixel, que, para vídeo

640x480 a 60 Hz, deve ser 25,175 MHz. Nessa configuração, segundo os padrões da indústria, as especificações de tempo da linha e do frame são dadas pelas tabelas:

TABELA 8 - TEMPORIZAÇÃO HORIZONTAL (LINHA)

PARTE DA LINHA	PIXELS	TEMPO [QS]
Visible area	640	25.422045680238
Front porch	16	0.63555114200596
Sync pulse	96	3.8133068520357
Back porch	48	1.9066534260179
Whole line	800	31.777557100298

TABELA 9 - TEMPORIZAÇÃO VERTICAL (FRAME)

PARTE DO FRAME	LINHAS	TEMPO [MS]
Visible area	480	15.253227408143
Front porch	10	0.31777557100298
Sync pulse	2	0.063555114200596
Back porch	33	1.0486593843098
Whole frame	525	16.683217477656

É importante salientar que esses valores são os utilizados para vídeo de resolução VGA, ou 640x480, com taxa de frame de 30 Hz. Portanto, para vídeo com características diferentes é necessário utilizar as especificações adequadas.

TABELA 10 - PINAGEM DO FPGA RELACIONADA À PORTA VGA

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
VGA_R[0]	A9	Bit 0 da componente vermelha - LSB
VGA_R[1]	C10	Bit 1 da componente vermelha
VGA_R[2]	A10	Bit 2 da componente vermelha
VGA_R[3]	B10	Bit 3 da componente vermelha - MSB
VGA_G[0]	C13	Bit 0 da componente verde - LSB
VGA_G[1]	A13	Bit 1 da componente verde
VGA_G[2]	B13	Bit 2 da componente verde
VGA_G[3]	A14	Bit 3 da componente verde - MSB
VGA_B[0]	B14	Bit 0 da componente azul - LSB
VGA_B[1]	A15	Bit 1 da componente azul
VGA_B[2]	B15	Bit 2 da componente azul
VGA_B[3]	A16	Bit 3 da componente azul - MSB
VGA_HS	B16	Sinal de sincronismo horizontal
VGA_VS	A17	Sinal de sincronismo vertical

REFERÊNCIA 7 - CONECTOR VGA

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
J3	Connector DB15HD Right Angle	-	A35116-ND

2.3.9 UTILIZANDO A PORTA RS-232

A interface serial RS-232 disponível na placa Mercurio® IV é interessante para estabelecer comunicação serial assíncrona entre a placa e outro dispositivo, que também possua uma porta serial RS-232.

Essa interface possui um componente na saída para que seja feita a conversão dos sinais digitais do FPGA para os sinais estabelecidos pelo padrão RS-232.

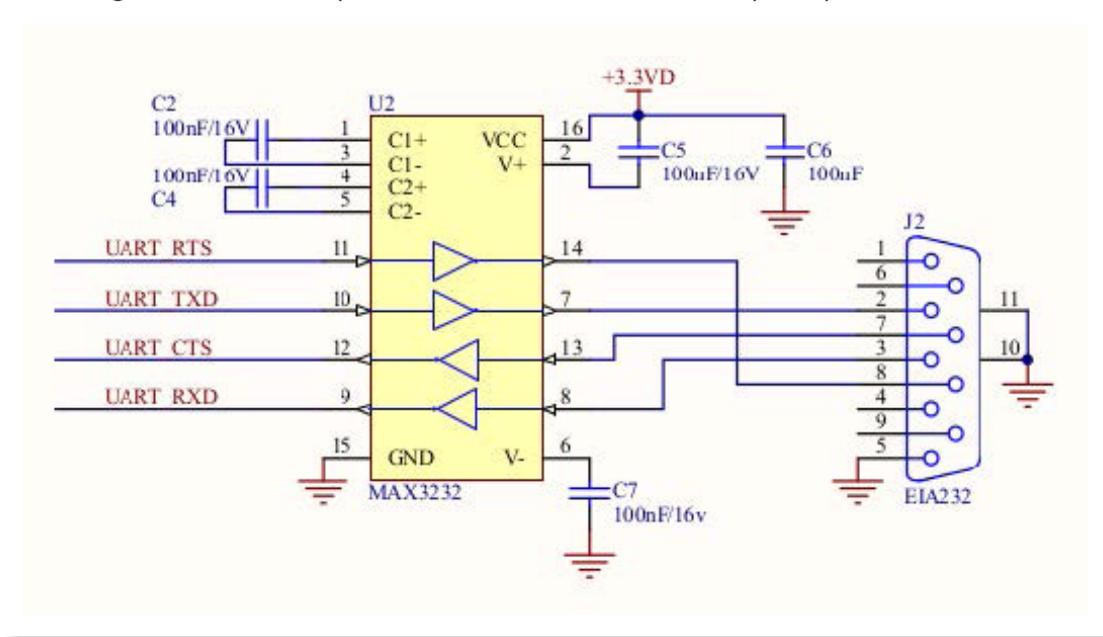


FIGURA 16 - COMPONENTE DE CONVERSÃO PARA RS-232

Na Mercurio® IV é possível utilizar handshake ou não, pois estão presentes os sinais Request to Send, pelo qual a porta de envio sinaliza para a de recepção que deseja enviar uma mensagem pela porta serial, e Clear to Send, pelo qual a porta de recepção sinaliza para a porta de envio que a linha está livre para o envio de dados. Porém também é possível não utilizar esses sinais e realizar a comunicação sem handshake, que é mais simples de ser implementada, porém é menos confiável. Portanto, a porta serial da Mercurio® IV pode operar nos modos 3-wire e 5-wire, sem handshake e com handshake, respectivamente.

A transmissão de dados deve obedecer o padrão, sendo necessários os start e stop bits. Ambos dispositivos devem ser configurados para o mesmo baud rate, ou taxa de transmissão de dados, número de bits da palavra, número de start e stop bits, utilização de bit de paridade e outros detalhes da comunicação.

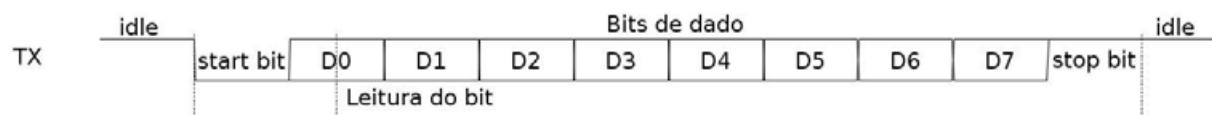


FIGURA 17 - FORMATO COMUMENTE UTILIZADO EM TRANSMISSÕES SERIAIS ASSÍNCRONAS

Para motivos de testes apenas, pode ser interessante colocar a porta serial da

Mercurio® IV em modo loopback, que consiste simplesmente em conectar os pinos, da própria porta serial da placa, UART_RTS com UART_CTS e UART_TXD com UART_RXD, assim, a placa irá se comunicar com ela mesma, sendo possível avaliar se o protocolo de comunicação está funcional.

TABELA 11 - PINAGEM DO FPGA RELACIONADA À PORTA SERIAL

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
UART_CTS	C7	Clear to send, indica que o dado pode ser enviado
UART_RTS	C6	Request to send, indica que quer enviar um dado
UART_RXD	C3	Pino de recepção de dados da UART
UART_TXD	C4	Pino de transmissão de dados da UART

REFERÊNCIA 8 - PORTA RS-232

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
J2	Connector DB9 Right Angle	-	DBPN1F-09
U2	3.0V TO 5.5V, Low-Power, up to 1Mbps, True RS-232 Transceiver Using Four 0.1QF External Capacitor	Maxim Integrated	MAX3232CUE

2.3.10 UTILIZANDO A PORTA USB

Além do conector para o USB-Blaster™, que é utilizado apenas para configuração e/ou alimentação da placa, a placa Mercurio® IV possui outra conexão USB, localizada na parte superior da placa, entre os conectores RS-232 e VGA. Essa conexão USB está configurada no modo Device, e portanto é destinada à comunicação entre o FPGA e periféricos externos com USB no modo Host, como um computador.

A porta USB é ligada ao FPGA através de um componente que gerencia toda a parte da comunicação USB, o FTDI FT245RQ, disponibilizando para o FPGA os dados na forma de uma FIFO.

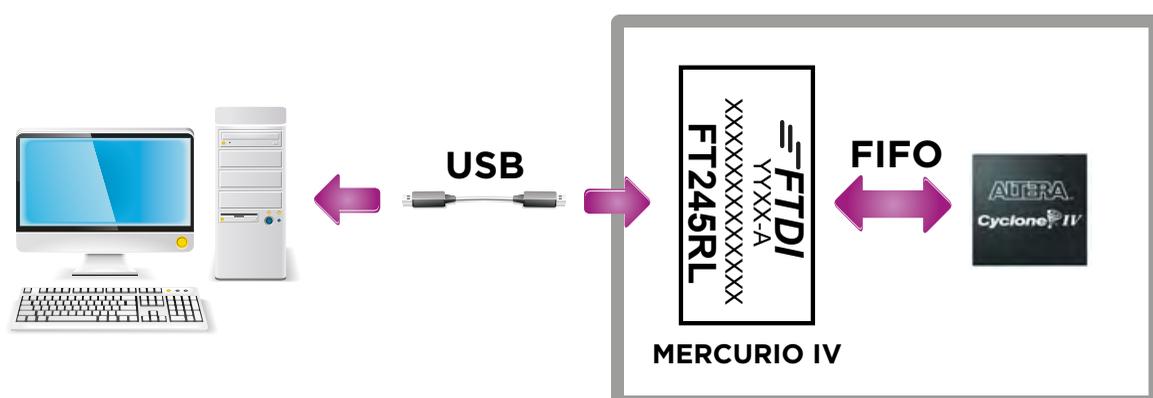


FIGURA 18 - FTDI ENTRE O CONECTOR USB E O BARRAMENTO DE ENTRADA DO FPGA NA PLACA MERCURIO® IV

Com a ajuda do componente FTDI FT245RQ, a utilização da USB é muito facilitada, pois não é necessário se preocupar com o protocolo da comunicação USB, e sim com o fluxo de dados da comunicação.

Para a comunicação entre um computador e a placa Mercurio® IV, a fabricante FTDI fornece um driver chamado Virtual Com Port, (VCP), que apresenta ao usuário a USB como um terminal serial. A FTDI também fornece drivers mais genéricos, facilitando no desenvolvimento de softwares de computador mais elaborados para realizar a comunicação com o FT245RQ. A partir de um computador, basta utilizar algum desses drivers disponíveis para enviar ou receber dados pela USB. As funções disponíveis e suas características são detalhadas pela fabricante FTDI, nos documentos relacionados aos seus drivers.

Já na comunicação entre FT245RQ e FPGA, no sentido FT245RQ ▶ FPGA, recepção de dados, o sinal RXF# sinaliza para o FPGA que existe um dado para ser lido da FIFO, e então o FPGA pode acionar o sinal RD# para fazer a leitura do byte.

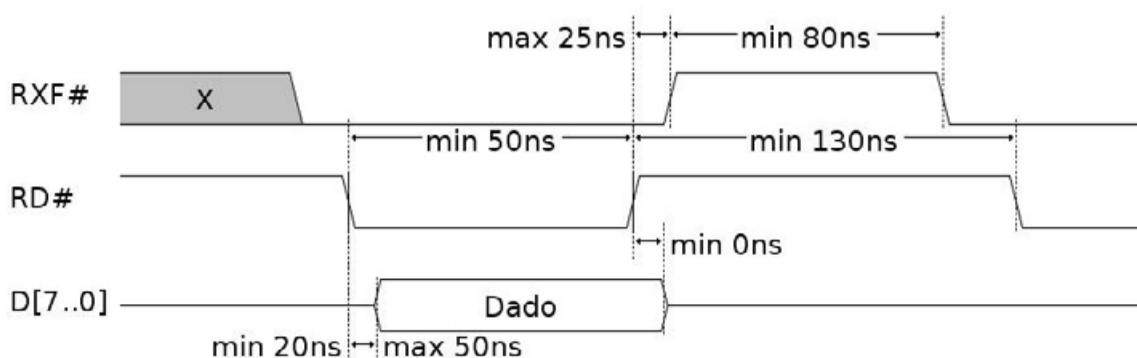


FIGURA 19 - SINAIS PARA LEITURA DE DADO DA FIFO DO FTDI FT245RQ

E no sentido FPGA ▶ FT245RQ, transmissão de dados, o sinal TXE# sinaliza para o FPGA quando é possível escrever um dado na FIFO, então o FPGA coloca o byte no barramento de dados e aciona o sinal WR, escrevendo o dado na FIFO do componente FT245RQ, que irá enviar o dado pela USB.

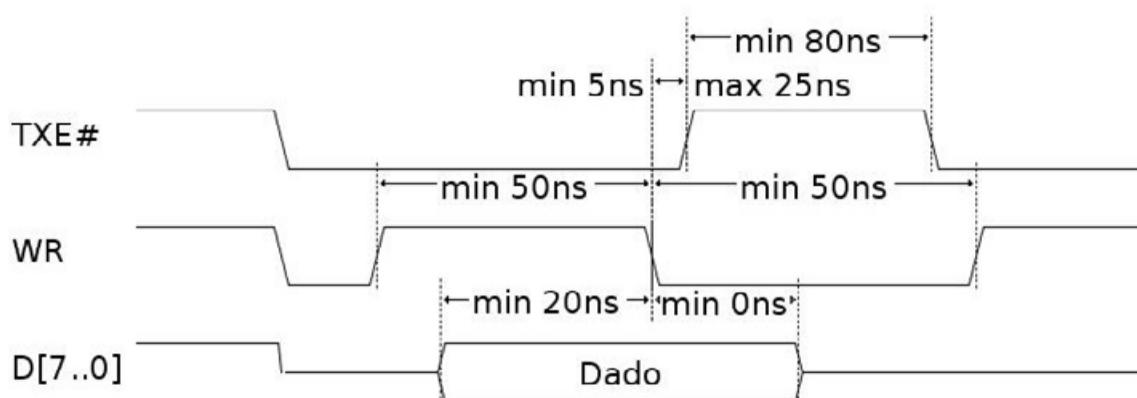


FIGURA 20 - SINAIS PARA ESCRITA DE DADO NA FIFO DO FTDI FT245RQ

As Figuras 19 e 20 foram baseadas no datasheet do componente FTDI FT245RQ.

TABELA 12 - PINAGEM DO FPGA RELACIONADA À PORTA USB

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
USB_PWREN_N	B8	Indica quando o componente está configurado e a interface USB pode ser utilizada
USB_RD_N	A8	Inicia uma leitura da FIFO
USB_RXF_N	A3	Indica que existe dado na FIFO
USB_TXE_N	B3	Indica que a FIFO tem espaço para um novo dado
USB_WR	B9	Inicia uma escrita na FIFO
USB_DATA[0]	B4	Bit 0 do barramento de dado - LSB
USB_DATA[1]	A5	Bit 1 do barramento de dado
USB_DATA[2]	B5	Bit 2 do barramento de dado
USB_DATA[3]	A7	Bit 3 do barramento de dado
USB_DATA[4]	A4	Bit 4 do barramento de dado
USB_DATA[5]	A6	Bit 5 do barramento de dado
USB_DATA[6]	B7	Bit 6 do barramento de dado
USB_DATA[7]	B6	Bit 7 do barramento de dado - MSB

REFERÊNCIA 9 - PORTA USB

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
J1	USB 2.0, Right Angle, SMT, A Type, Receptacle, 5 Position, Black	-	MSB5BSW-X-H- CN
U4	USB/Parallel FIFO Bi-Directional Data Transfer Chip, QFN32, Pb-Free	FTDI Chip	FT245RQ-REEL

2.3.11 UTILIZANDO O CONVERSADOR AD

O conversor analógico-digital do kit Mercurio IV possui duas entradas analógicas diferenciais, ou 4 entradas single ended, dois canais de conversão paralelos e duas saídas digitais seriais. O conversor AD possui resolução de 12 bits e se comunica com o FPGA Cyclone IV através de um barramento serial SPI, que opera nos modos 0 ou 3. O conversor não possui nenhuma entrada no barramento SPI, apenas as saídas relativas a cada canal de conversão, então, o controle do conversor é feito através de alguns sinais digitais de entrada, como ADC_CNVST e ADC_SD.

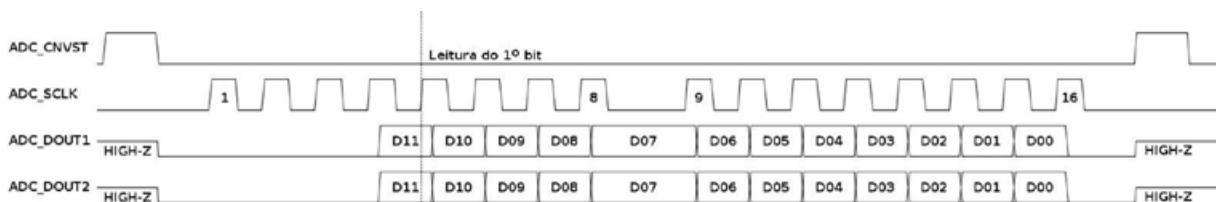


FIGURA 21 - FORMA DE ONDA DO CONVERSADOR AD NO MODO DUAL

Para uma conversão ser iniciada, o sinal ADC_CNVST deve ser colocado em nível lógico baixo (sensível a borda de descida), enquanto o ADC_SCL está em idle. Feito isso, o estágio de entrada do conversor segura o valor analógico atual e começa a conversão. A conversão é feita com base no clock ADC_SCL.

Após o início da conversão, existe uma latência de 3 pulsos do sinal ADC_SCL para as saídas começarem a ser atualizadas, assim, o resultado da conversão somente começa a ser enviado na saída do conversor após a quarta borda de subida do sinal ADC_SCL, o que significa que a leitura do primeiro bit do resultado da conversão é feita apenas na quinta borda de subida do sinal ADC_SCL, após a borda de descida de ADC_CNVST.

O conversor AD, por possuir dois canais, pode trabalhar no modo dual, ADC_SD em nível lógico baixo, ou então no modo single, ADC_SD em nível lógico alto. No modo dual, os resultados das conversões dos dois canais são enviados para as saídas ADC_DOUT[1] e ADC_DOUT[2] paralelamente. Já no modo single, somente uma saída é utilizada, a ADC_DOUT[1], então, o resultado da conversão do canal 2 é enviado para a saída somente após todo o canal 1 ter sido enviado.

Assim, o desempenho do conversor depende do modo utilizado. No modo dual são necessários 16 pulsos de ADC_SCL (4 de latência inicial mais 12 para os resultados das conversões serem enviados), já no modo single são necessários 28 pulsos de ADC_SCL (4 de latência inicial mais 24 de dado). Isso significa que no modo dual, a capacidade do ADC é de 1,25 Msps, já no modo single é de 0,625 Msps (considerando a máxima frequência de ADC_SCL). Portanto, é recomendada a operação do conversor AD em modo dual, quando necessário o uso de 2 canais.

É possível aproveitar as 4 entradas analógicas do conversor. Para isso, inicialmente duas entradas são convertidas (modo dual), em seguida a seleção do MUX, ADC_SEL, deve ser mudada para as outras duas entradas, para então essas serem convertidas. Assim, ao término dessas duas etapas, as 4 entradas analógicas terão sido convertidas.

TABELA 13 - PINAGEM DO FPGA RELACIONADA AO CONVERSOR DA

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
ADC_CNVST	AB18	Sinaliza que uma nova conversão deve ser iniciada
ADC_CS_N	AA17	Chip Select
ADC_DOUT[0]	AB19	Saída para conversão no modo single ou para o canal 1 em modo dual
ADC_DOUT[1]	AA19	Saída para o canal 2 em modo dual
ADC_REFSEL	AB17	Define qual será a tensão de referência para ADC
ADC_SCLK	AA18	Clock da comunicação serial
ADC_SD	AA16	Seleção entre os modos single ou dual
ADC_UB	AB20	Seleção entre modo unipolar ou modo bipolar (entradas diferenciais)
ADC_SEL	AB16	Seleciona se serão utilizadas as entradas analógicas primárias ou secundárias (modo unipolar apenas)

REFERÊNCIA 10 - CONVERSOR DA

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
J8	3-Conductor Jack with 2 break contacts (normals) and 2 auxiliary make contacts	-	SJ1-3535NG-BE
U16	Dual, 12-Bit, 1.25Msps, Simultaneous-Sampling ADCs with Serial Interface	Maxim Integrated	MAX1379

2.3.12 UTILIZANDO O CONVERSOR DA

O conversor digital-analógico possui dois canais de conversão e duas saídas analógicas, porém possui apenas uma entrada serial, pela qual os dois buffers de conversão são alimentados e todas as configurações do conversão são realizadas. A única entrada digital que não faz parte do barramento SPI é um sinal de clear assíncrono, DAC_CLR_N, que limpa os buffers de conversão, fazendo as duas saídas irem para zero (nível mínimo de tensão).

A comunicação SPI do conversor DA pode ser nos modos 1 ou 2, como é possível ver na Figura 22. Todo comando enviado para o conversor tem 24 bits de comprimento, portanto, após o sinal DAC_CS_N ir para nível lógico baixo, os 24 bits devem ser enviados pelo pino DAC_DIN que será registrado nas bordas de descida de DAC_SCLK.

A saída analógica é resultado da conversão do valor armazenado no registrador CODE. O valor digital é armazenado no registrador utilizando o comando CODEn.

O comando CODEn armazena o valor digital no registrador CODE, e então o comando LOADn transfere o valor do registrador CODE para o registrador DAC, iniciando a conversão DA. É possível utilizar o comando CODEn_LOADn, que automaticamente escreve o valor digital no registrador CODE e o transfere para o registrador DAC para iniciar a conversão. É possível configurar o conversor para que a transferência do dado no registrador CODE para o registrador DAC seja transparente, ou seja, basta realizar o comando CODEn que a conversão será iniciada.

TABELA 14 - COMANDOS DE ESCRITA DO DAC

COMANDO	B23	B22	B21	B20	B19	B18	B17	B16	B15	B14	B13	B12	B11	B10	B09	B08	B07	B06	B05	B04	B03	B02	B01	B00	DESCRIÇÃO	
CODEn	0	0	0	0	Seleção do DAC				Registrador CODE[1..4]				Registrador CODE[3..0]				X	X	X	X	X	X	X	X	X	Escreve no registrador CODE
LOADn	0	0	0	1	Seleção do DAC				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	Transfere do registrador CODE para o registrador DAC
CODEn-LOADn	0	0	1	1	Seleção do DAC				Registrador CODE[1..4]				Registrador CODE[3..0]				X	X	X	X	X	X	X	X	X	Escreve no registrador CODE e transfere para o registrador DAC

TABELA 15 - CONFIGURAÇÃO DO DAC

COMANDO	B23	B22	B21	B20	B19	B18	B17	B16	B15	B14	B13	B12	B11	B10	B09	B08	B07	B06	B05	B04	B03	B02	B01	B00	DESCRIÇÃO	
CONFIG	0	1	1	0	0	0	0	LD_EN	X	X	X	X	X	X	DAC A	DAC B	X	X	X	X	X	X	X	X	X	Configura transparência

As configurações de funcionamento do conversor são feitas de forma similar à escrita do registrador CODE, mudando-se o código do comando. A Tabela 15 mostra o formato do comando a ser enviado via SPI para configurar a transparência citada anteriormente. Nessa configuração, se LD_EN é 1, a escrita do registrador CODE com o comando CODEn automaticamente alimenta o registrador DAC e inicia a conversão.

Dessa forma, também é possível selecionar a referência de tensão para o DAC, desligar o conversor (as saídas vão para zero), limpar os registradores de conversão (clear por software) e resetar todas as configurações e registradores do conversor (reset por software). Ao todo são 12 comandos possíveis, todos descritos no datasheet do conversor.

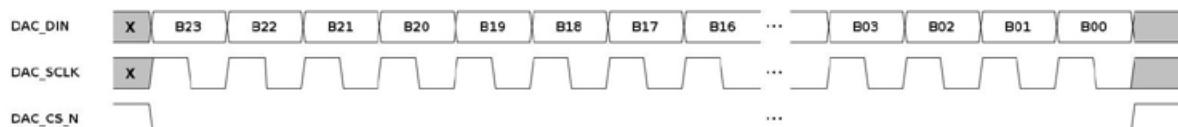


FIGURA 22 - FORMA DE ONDA DO CONVERSOR DA

TABELA 16 - PINAGEM DO FPGA RELACIONADA AO CONVERSOR DA

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
DAC_CLR_N	AA15	Limpa os buffers do conversor, colocando as saídas em zero
DAC_CS_N	AB15	Chip Select
DAC_DIN	AB14	Entrada de dado da comunicação serial
DAC_SCLK	AA14	Clock da comunicação serial

REFERÊNCIA 11 - CONVERSOR DA

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
J7	3-Conductor Jack with 2 break contacts (normals) and 2 auxiliary make contacts	-	SJ1-3535NG-GR
U13	Ultra-Small, Dual-Channel, 12-Bit Buffered Output DACs with Internal Reference and SPI Interface	Maxim Integrated	MAX5702AAUB+

2.3.13 UTILIZANDO O CARTÃO MICROSD

O cartão MicroSD pode ser utilizado como uma expansão de memória da placa Mercurio® IV. O MicroSD foi adotado pela sua popularidade e robustez, sendo assim uma memória de fácil acesso, com garantias de funcionamento.

O FPGA está ligado ao cartão MicroSD por sete sinais, quatro sinais de dado, dois sinais de controle e um sinal de clock. Como o FPGA tem acesso aos quatro sinais de dado do MicroSD, a interface entre o FPGA e o cartão pode ser feita nos seguintes modos:

- SPI – comunicação mais simples e genérica, porém velocidade de transmissão baixa. Para essa configuração, os pinos do FPGA devem obedecer a seguinte relação:

- SD_DATA(3) ▶ SPI_CS;
- SD_DATA(0) ▶ SPI_MISO;
- SD_CMD ▶ SPI_MOSI;
- SD_CLK ▶ SPI_SCK

- SD mode, de 1 até 4 bits – protocolo um pouco mais complexo, porém alcançando velocidades maiores que no modo SPI, principalmente na comunicação com 4 bits.

É importante lembrar que, caso seja necessário que o cartão MicroSD também possa ser acessado por um computador, deve ser utilizado um sistema de arquivos compatível com o sistema operacional do computador, e que por isso, é recomendado utilizar um microprocessador, como o NIOS®, para fazer o gerenciamento do sistema de arquivos e acessar corretamente o cartão MicroSD. Existe uma biblioteca, compatível com ANSI C, que implementa funções relacionadas ao sistema de arquivos FAT, ela pode ser utilizada em sistemas embarcados para o acesso à cartões SD. O nome dessa biblioteca é FatFS Generic FAT File System Module.

Caso o cartão de memória seja apenas utilizado na placa Mercurio® IV, o sistema de arquivos não é necessário, pois o usuário pode desejar apenas escrever bytes sequencialmente na memória, para uso apenas na placa Mercurio® IV.

TABELA 17 - PINAGEM DO FPGA RELACIONADA AO CARTÃO MICROSD

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO MODO SD	DESCRIÇÃO SPI
SD_CLK	Y7	Clock da interface com o cartão MicroSD	Clock serial
SD_CD_N	U7	Pino de identificação se o cartão está inserido	NC
SC_CMD	W7	Sinal de comando, indica que uma operação será iniciada	MOSI
SD_D[0]	W6	Bit 0 do barramento de operação/dado - LSB	MISO
SD_D[1]	Y6	Bit 1 do barramento de operação/dado	NC
SD_D[2]	W8	Bit 2 do barramento de operação/dado	NC
SD_D[3]	Y8	Bit 3 do barramento de operação/dado - MSB	CS

REFERÊNCIA 12- CONECTOR MICROSD

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
J5	Micro-SDcard PushPull connector	-	DM3D-SF

2.3.14 UTILIZANDO A MEMÓRIA FLASH

A memória Flash EPCS do kit tem 64 Mbit (8M x 8) de capacidade, e possui interface de comunicação SPI, suportando até 25 MHz para escrita e 40 MHz para leitura. Essa memória tem capacidade para escritas e leituras sequenciais, é possível apagar a memória toda ou apenas blocos da memória (o bit, quando apagado, fica com valor '1') e existem mecanismos de proteção de setores da memória. O Byte de status guarda as informações sobre a proteção dos setores da memória, além de possuir um bit que indica quando a memória está sendo escrita, ou apagada.

Essa memória é organizada em:

- 8388608 bytes, 8 bits cada;
- 128 setores, 65536 Bytes cada;
- 32768 páginas, 256 Bytes cada.

Cada página pode ser escrita de uma vez, utilizando-se a escrita sequencial, porém para escrever mais de uma página é necessário realizar uma nova escrita, indicando o endereço da nova página. A proteção da memória é feita por grupos de setores, definidos no registrador de status. Por fim, cada setor pode ser apagado de cada vez, independentemente, mas também é possível apagar a memória toda de uma só vez.

A comunicação com a memória é serial SPI, modos 0 ou 3, e os comandos possíveis são listados na Tabela 18. O datasheet da memória detalha o funcionamento e a utilização de cada um desses comandos.

TABELA 18 - COMANDOS SPI DA MEMÓRIA FLASH

COMANDO	CÓDIGO	BYTES DE ENDEREÇO	BYTES DUMMY	BYTES DE DADO	DAC_SLCK FMAX
Write enable	0000 0110	0	0	0	25
Write disable	0000 0100	0	0	0	25
Read status	0000 0101	0	0	1 - infinito	25
Read Bytes	0000 0011	3	0	1 - infinito	20
Read ID	1010 1011	0	3	1 - infinito	25
Fast read	0000 1011	3	1	1 - infinito	40
Write status	0000 0001	0	0	1	25
Write Bytes	0000 0010	3	0	1 - 256	25
Erase bulk	1100 0111	0	0	0	25
Erase sector	1101 1000	3	0	0	25

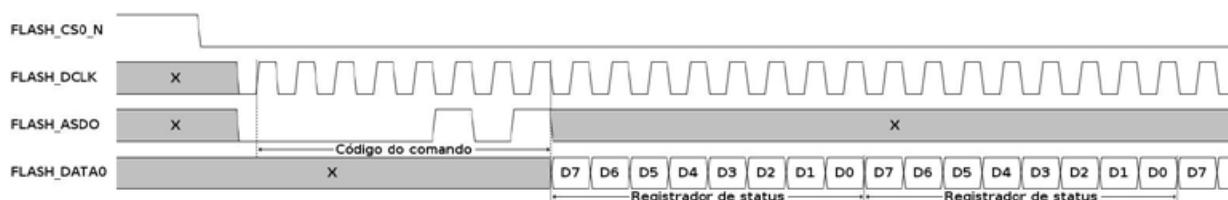


FIGURA 23 - FORMA DE ONDA DA LEITURA DO REGISTRADOR DE STATUS DA FLASH

Além de ser utilizada como memória de armazenamento da configuração do FPGA, essa memória Flash também pode ser utilizada como memória não volátil de usuário. Dessa forma, dados relevantes para a aplicação podem ser armazenados na Flash para não serem perdidos quando a placa for desligada.

É importante ressaltar que essa memória é também utilizada como memória EPCS (memória de configuração do FPGA) do kit Mercurio® IV, portanto, os primeiros endereços da memória sempre serão utilizados para dados de configuração do FPGA. Assim, é muito importante que esses dados não sejam sobrescritos ou apagados da memória, para garantir o bom funcionamento do FPGA. Para evitar qualquer problema, pode-se proteger a área da memória com dados de configuração do FPGA contra escritas indesejadas.

Cabe ao usuário realizar o gerenciamento dessa memória, a fim de não comprometer os dados de configuração.

TABELA 19 - PINAGEM DO FPGA RELACIONADA À MEMÓRIA FLASH

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
FLASH_DCLK	K2	Clock da interface SPI com a memória Flash
FLASH_DATA0	K1	Sinal de dados de saída da Flash (entrada do FPGA)
FLASH_ASDO	D1	Sinal de dados de entrada da Flash (saída do FPGA)
FLASH_CS0_N	E2	Chip Select da memória Flash

REFERÊNCIA 13 - MEMÓRIA FLASH

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
U6	64 Mbit, low voltage, Serial Flash memory	Altera	EPCS64SI16N

2.3.15 UTILIZANDO A MEMÓRIA SDRAM

A memória SDRAM é a memória de alta velocidade do kit Mercurio® IV. Ela pode ser utilizada para o processamento de dados, quando existe um stream de dados, como, por exemplo, um vídeo. Além dessa característica, ela é uma memória de alta capacidade, podendo armazenar muitos Bytes de dados. É importante salientar que a SDRAM é uma memória volátil, então, diferente da memória Flash, ela perde os dados quando a alimentação de energia é desligada, e além disso, durante a

utilização, é necessário fazer o refresh da memória num período de tempo.

A memória SDRAM disponível na placa Mercurio® IV possui 512 Mbits. São quatro bancos de memória, cada um possui 8 M x 16 bits. Cada banco é estruturado em 8192 linhas e 1024 colunas de 16 bits.

O acesso aos bancos de memória é feito através dos seletores de banco de memória, SDRAM_BA[0] e SDRAM_BA[1], o endereço de memória (linha ou coluna dentro do banco selecionado) é colocado nos pinos SDRAM_A[12..0] e o dado de 16 bits é inserido nos pinos SDRAM_D[15..0].

Ainda existem os sinais de controle SDRAM_CKE que habilita ou desabilita o clock na memória, SDRAM_CS_N que habilita ou desabilita o funcionamento de todas as entradas da memória, exceto SDRAM_CKE, e os bits SDRAM_DQMH e SDRAM_DQML habilitam ou desabilitam cada um dos bytes de dado, ou seja, se o MSB é desativado numa escrita, somente o LSB é escrito, e vice versa.

Enfim, para realizar operações na memória são utilizados os bits de comando SDRAM_RAS_N (operação em uma linha), SDRAM_CAS_N (operação em uma coluna) e SDRAM_WE_N (operação de escrita ou leitura). A combinação deles gera diferentes comandos, os quais devem ser utilizados em determinada ordem para realizar a operação desejada. Algumas operações que podem ser feitas sobre a memória SDRAM são listadas na Tabela 20.

Para iniciar uma operação a função Ativar deve ser executada, essa função abre a linha escolhida do banco escolhido. Uma linha aberta está disponível para leitura ou escrita. Após o comando Ativar, podem ser enviados os comandos Ler ou Escrever, nos quais a coluna inicial deve ser informada, assim, é iniciado um burst de leitura ou escrita na memória. Quando for desejado interromper um burst de leitura ou escrita, basta enviar o comando Terminar Burst, que irá terminar a operação de leitura ou escrita e irá fechar a linha. O tamanho do burst é definido nos registradores de configuração da memória, com o comando Configurar Memória

TABELA 20 - ALGUMAS OPERAÇÕES SOBRE A MEMÓRIA SDRAM

NOME (FUNÇÃO)	NCS	NRAS	NCAS	NWE	DQM	ADDR	DQ
Entradas desativadas	H	X	X	X	X	X	X
Sem operação	L	H	H	H	X	X	X
Ativar	L	L	H	H	X	Banco/ linha	X
Ler	L	H	L	H	L/H	Banco/ coluna	X
Escrever	L	H	L	L	L/H	Banco/ coluna	Dado
Terminar Burst	L	H	H	L	X	X	Ativo
Configurar Memória	L	L	L	L	X	Opcode	X

A Figura 24 mostra a forma de onda do comando Ativar. As demais funções são detalhadas no datasheet da memória, portanto é altamente recomendada a leitura desse datasheet quando necessário trabalhar com a memória. Além disso, é importante atentar para os requisitos de temporização da memória, por exemplo, a latência na leitura de dados.

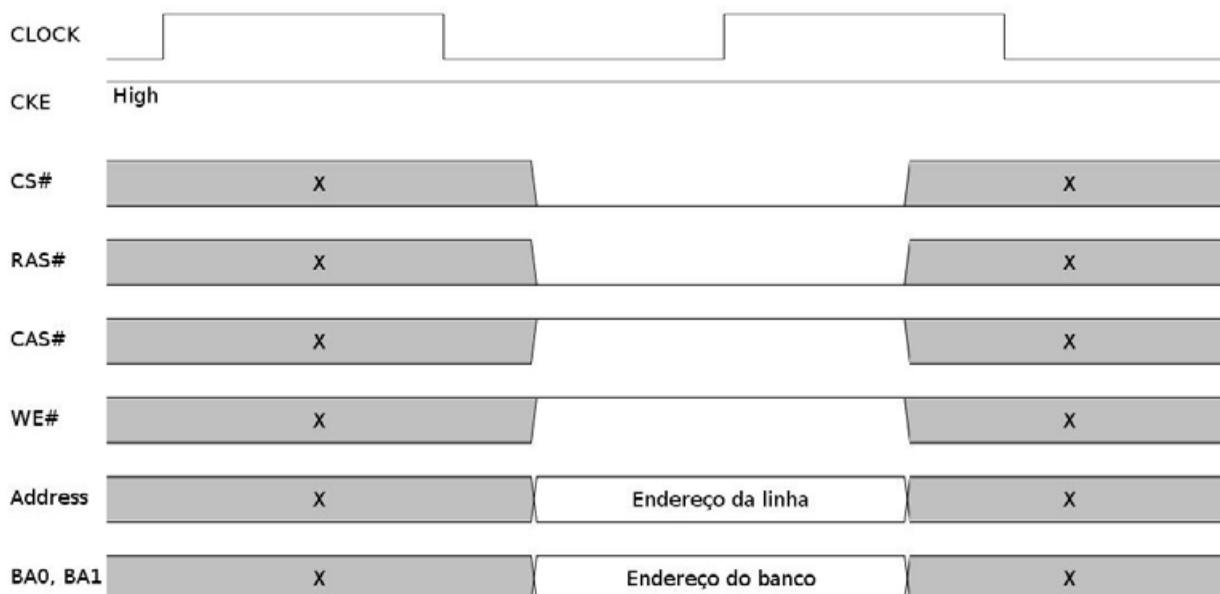


FIGURA 24 - FORMA DE ONDA DO COMANDO ATIVAR

Como são 8192 linhas em cada banco, quando for feito um comando Ativar (que ativa uma linha), os 13 bits de endereçamento devem ser escritos, já quando for um comando Ler (ou Escrever, comandos que atuam sobre uma parte da linha ativa, a partir de determinada coluna), apenas 10 bits de endereçamento precisam ser utilizados. No caso de se realizar um comando sobre uma coluna, o bit 10 do endereço, SDRAM_A[10], indica se o fechamento da linha ativa deve ser feito após a operação ou não.

TABELA 21 - PINAGEM DO FPGA RELACIONADA À MEMÓRIA SDRAM

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
SDRAM_A[0]	P2	Bit 0 de endereço - LSB
SDRAM_A[1]	R1	Bit 1 de endereço
SDRAM_A[2]	P3	Bit 2 de endereço
SDRAM_A[3]	R2	Bit 3 de endereço
SDRAM_A[4]	P4	Bit 4 de endereço
SDRAM_A[5]	P5	Bit 5 de endereço
SDRAM_A[6]	N5	Bit 6 de endereço
SDRAM_A[7]	M4	Bit 7 de endereço
SDRAM_A[8]	N1	Bit 8 de endereço
SDRAM_A[9]	M2	Bit 9 de endereço
SDRAM_A[10]	P1	Bit 10 de endereço
SDRAM_A[11]	M3	Bit 11 de endereço
SDRAM_A[12]	L6	Bit 12 de endereço - MSB
SDRAM_BA[0]	J4	Bit 0 do endereço de seleção de Banco - LSB
SDRAM_BA[1]	H2	Bit 1 do endereço de seleção de Banco - MSB

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
SDRAM_D[0]	B2	Bit 0 de dado - LSB
SDRAM_D[1]	B1	Bit 1 de dado
SDRAM_D[2]	C2	Bit 2 de dado
SDRAM_D[3]	C1	Bit 3 de dado
SDRAM_D[4]	D2	Bit 4 de dado
SDRAM_D[5]	E3	Bit 5 de dado
SDRAM_D[6]	F2	Bit 6 de dado
SDRAM_D[7]	F1	Bit 7 de dado
SDRAM_D[8]	J2	Bit 8 de dado
SDRAM_D[9]	H6	Bit 9 de dado
SDRAM_D[10]	H5	Bit 10 de dado
SDRAM_D[11]	J3	Bit 11 de dado
SDRAM_D[12]	G3	Bit 12 de dado
SDRAM_D[13]	G4	Bit 13 de dado
SDRAM_D[14]	G5	Bit 14 de dado
SDRAM_D[15]	E4	Bit 15 de dado - MSB
SDRAM_DQM[0]	E1	Byte Enable do dado (habilita/desabilita os bits 0 até 7)
SDRAM_DQM[1]	J5	Byte Enable do dado (habilita/desabilita os bits 8 até 15)
SDRAM_CLK	E5	Entrada de clock da memória
SDRAM_CKE	M5	Sinal de Clock Enable da memória
SDRAM_CAS_N	M1	Column Addr Strobe (utilizado para formar um comando)
SDRAM_CS_N	H1	Sinal de Chip Select da memória
SDRAM_RAS_N	N2	Row Addr Strobe(utilizado para formar um comando)
SDRAM_WE_N	J1	Write Enable (utilizado para formar um comando)

REFERÊNCIA 14 - MEMÓRIA SDRAM

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
U9	IC SDRAM 512MBIT 143MHz, 54TSOP	ISSI	IS42S16320B

2.3.16 UTILIZANDO A INTERFACE ETHERNET

A placa Mercurio® IV também suporta comunicação via Ethernet. Na placa foi instalado um componente PHY Ethernet, que realiza as funções necessárias à camada física da comunicação Ethernet. É esse componente que recebe a informação do MAC contido no FPGA e envia pela rede e, também, recebe informação da rede e disponibiliza para o MAC do FPGA. O PHY da placa Mercurio IV funciona nos formatos Ethernet 10BASE-T e 100BASE-TX, podendo trabalhar com transmissões de até 100 Mbit/s.

O MAC é o responsável por interpretar as mensagens recebidas e montar as mensagens para envio pela Ethernet, assim, as camadas superiores ao MAC precisam

apenas se preocupar com o fluxo de dados, transmitidos e recebidos. Recomenda-se a utilização do IP da Altera chamado Triple-Speed Ethernet para realizar essa função, assim o desenvolvedor apenas se preocupa com as camadas superiores.

Caso a comunicação seja baseada no protocolo TCP/IP, o FPGA deve suportar também esse protocolo. Para isso recomenda-se a utilização do microprocessador softcore NIOS® II, em conjunto com as funções que implementam a pilha TCP. Com uma implementação assim, é possível fazer a placa Mercurio® IV funcionar como um servidor web, por exemplo, exibindo uma página em HTML para o usuário.

Já caso a comunicação seja baseada num protocolo customizado (entre dois kits Mercurio® IV, por exemplo), a interface Ethernet também pode ser utilizada de forma mais simples que a feita com o protocolo TCP/IP, mas o MAC ainda deverá ser utilizado.

TABELA 22 - PINAGEM DO FPGA RELACIONADA À INTERFACE ETHERNET

NOME DO SINAL	PINO DO FPGA	DESCRIÇÃO
ETH_RXD[0]	A20	Bit 0 do dado de recepção - LSB
ETH_RXD[1]	B19	Bit 1 do dado de recepção
ETH_RXD[2]	A19	Bit 2 do dado de recepção
ETH_RXD[3]	B18	Bit 3 do dado de recepção - MSB
ETH_TXD[0]	D21	Bit 0 do dado de transmissão - LSB
ETH_TXD[1]	D22	Bit 1 do dado de transmissão
ETH_TXD[2]	E21	Bit 2 do dado de transmissão
ETH_TXD[3]	E22	Bit 3 do dado de transmissão - MSB
ETH_RXDV	B20	Dado válido de recepção
ETH_RXER	B21	Erro de recepção
ETH_TXCLK	C21	Clock de transmissão
ETH_RXCLK	C20	Clock de recepção
ETH_RST_N	G11	Reset
ETH_MDIO	B17	Sinal bidirecional de dados de gerenciamento serial do PHY
ETH_MDC	A18	Sinal de clock de gerenciamento serial do PHY
ETH_CRS	F21	Deteção de portadora (se o meio está sendo utilizado)
ETH_COL	F22	Deteção de colisão
ETH_TXEN	C22	Habilita a transmissão do dado em ETH_TX
ETH_TXER	B22	Erro de transmissão

REFERÊNCIA 15 - PHY ETHERNET

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
U3	Connector RJ45 with internal magnetics	-	SI-60062-F
U7	MicroPHYTM 10/100BASE-TX Transceiver	Teridian	78Q2123

2.3.17 UTILIZANDO AS INTERFACES PMOD®

As interfaces PMOD® são utilizadas para conectar Peripheral Modules a uma placa host, nesse caso, a Mercurio® IV. Peripheral Modules são sistemas externos, com diferentes objetivos, e fabricados por diversos fabricantes, por exemplo, a Maxim possui uma linha desses dispositivos, que pode ser vista no site <http://www.maximintegrated.com/products/evkits/fpga-modules/>. Existem sensores de temperatura, proximidade, conversores DAC e ADC, entre outros, e é através da interface PMOD® que esses sistemas se comunicam com o FPGA da placa Mercurio® IV. Os módulos PMOD® trabalham em baixa frequência e possuem poucos pinos de IO. Existem versões dessa interface de 6 e de 12 pinos, sendo que a de 12 pinos está arranjada de forma a equivaler a duas de 6 pinos juntas. Cada interface de 6 pinos possui 4 pinos de dados, 1 pino de alimentação e 1 pino de terra (a interface de 12 pinos possui 8 pinos de dados, 2 de alimentação e 2 de terra).

A placa Mercurio® IV possui duas interfaces PMOD® de 12 pinos, portanto, até quatro módulos podem ser conectados à placa, contanto que os quatro sejam de 6 pinos cada.

A tensão de operação de barramentos PMOD® não é especificada, porém na placa Mercurio® IV é possível trabalhar com 3.3V ou 5V. Para fazer a seleção da tensão de operação é necessário posicionar os jumpers, situados ao lado de cada conector, corretamente de acordo com a tensão de operação.

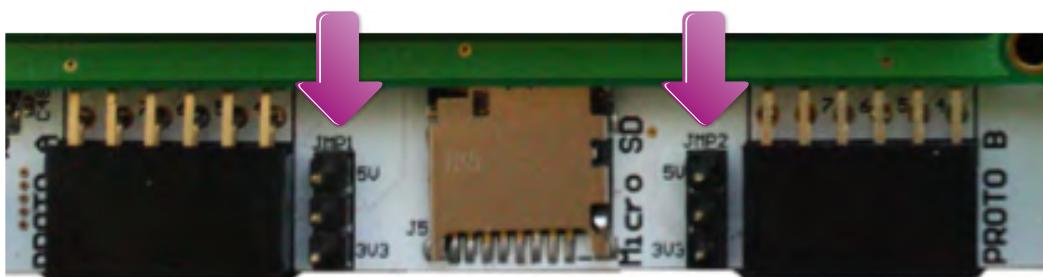


FIGURA 25 - JUMPERS DE SELEÇÃO DE TENSÃO

Atenção para não deixar o jumper em 5V e conectar um dispositivo POMOD de 3.3V, pois isso pode danificar o dispositivo. Recomenda-se, sempre que a placa não for mais utilizada, deixar os jumpers na posição 3V3, evitando danificar componentes por excesso de tensão.

A Figura 26 mostra o esquema do conector PMOD® disponível na placa Mercurio® IV e a função de cada pino do conector, lembrando as linhas são iguais. Cada linha pode ser utilizada para um módulo de 6 pinos e as duas linhas em conjunto podem ser utilizadas para um módulo de 12 pinos.

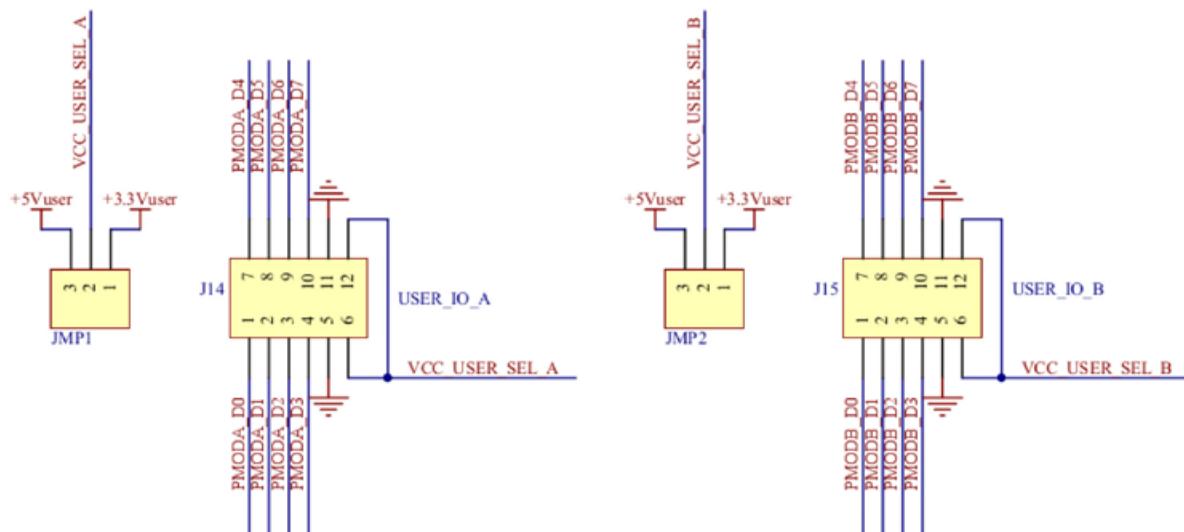


FIGURA 26 - INTERFACES PMOD® E FUNÇÕES DOS PINO

TABELA 23 - PINAGEM DO FPGA RELACIONADA ÀS INTERFACES PMOD®

NOME DO SINAL	PINO DO FPGA	PINO NO PMOD®	DESCRIÇÃO
INTERFACE A			
PROTO_A[0]	AB5	1	Pino 0 da interface A compatível com PMOD®
PROTO_A[1]	AB4	2	Pino 1 da interface A compatível com PMOD®
PROTO_A[2]	AB3	3	Pino 2 da interface A compatível com PMOD®
PROTO_A[3]	AA1	4	Pino 3 da interface A compatível com PMOD®
GND	-	5	GND
VCC	-	6	VCC de 3.3V ou 5V, dependendo da posição do jumper
PROTO_A[4]	AA7	7	Pino 4 da interface A compatível com PMOD®
PROTO_A[5]	AA5	8	Pino 5 da interface A compatível com PMOD®
PROTO_A[6]	AA4	9	Pino 6 da interface A compatível com PMOD®
PROTO_A[7]	AA3	10	Pino 7 da interface A compatível com PMOD®
GND	-	11	GND
VCC	-	12	VCC de 3.3V ou 5V, dependendo da posição do jumper

NOME DO SINAL	PINO DO FPGA	PINO NO PMOD®	DESCRIÇÃO
INTERFACE B			
PROTO_B[0]	AB10	1	Pino 0 da interface B compatível com PMOD®
PROTO_B[1]	AB9	2	Pino 1 da interface B compatível com PMOD®
PROTO_B[2]	AB8	3	Pino 2 da interface B compatível com PMOD®
PROTO_B[3]	AB7	4	Pino 3 da interface B compatível com PMOD®
GND	-	5	GND
VCC	-	6	VCC de 3.3V ou 5V, dependendo da posição do jumper
PROTO_B[4]	Y10	7	Pino 4 da interface B compatível com PMOD®
PROTO_B[5]	AA10	8	Pino 5 da interface B compatível com PMOD®
PROTO_B[6]	AA9	9	Pino 6 da interface B compatível com PMOD®
PROTO_B[7]	AA8	10	Pino 7 da interface B compatível com PMOD®
GND	-	11	GND
VCC	-	12	VCC de 3.3V ou 5V, dependendo da posição do jumper

REFERÊNCIA 16 - INTERFACE PMOD®

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
J14 e J15	Header, 6-Pin, Dual row	-	MND-06
JMP1 e JMP2	Header, 3-Pin	-	BPSC-MM-3 com JC-02

2.3.18 UTILIZANDO AS INTERFACES GPIO

Caso seja necessário conectar uma placa de expansão à placa Mercurio® IV, foram disponibilizados dois barramentos de pinos de propósito geral. Cada barramento possui 32 pinos de IO, 2 pinos de entrada de clock e 2 de saída de clock.

As placas de expansão, assim como os módulos PMOD®, possuem diferentes objetivos e funcionamentos. Diversos fabricantes também fornecem placas de expansão compatíveis com o barramento GPIO disponível na placa Mercurio® IV.

Os 32 pinos de dado podem ser configurados independentemente como entrada, saída ou bidirecional.

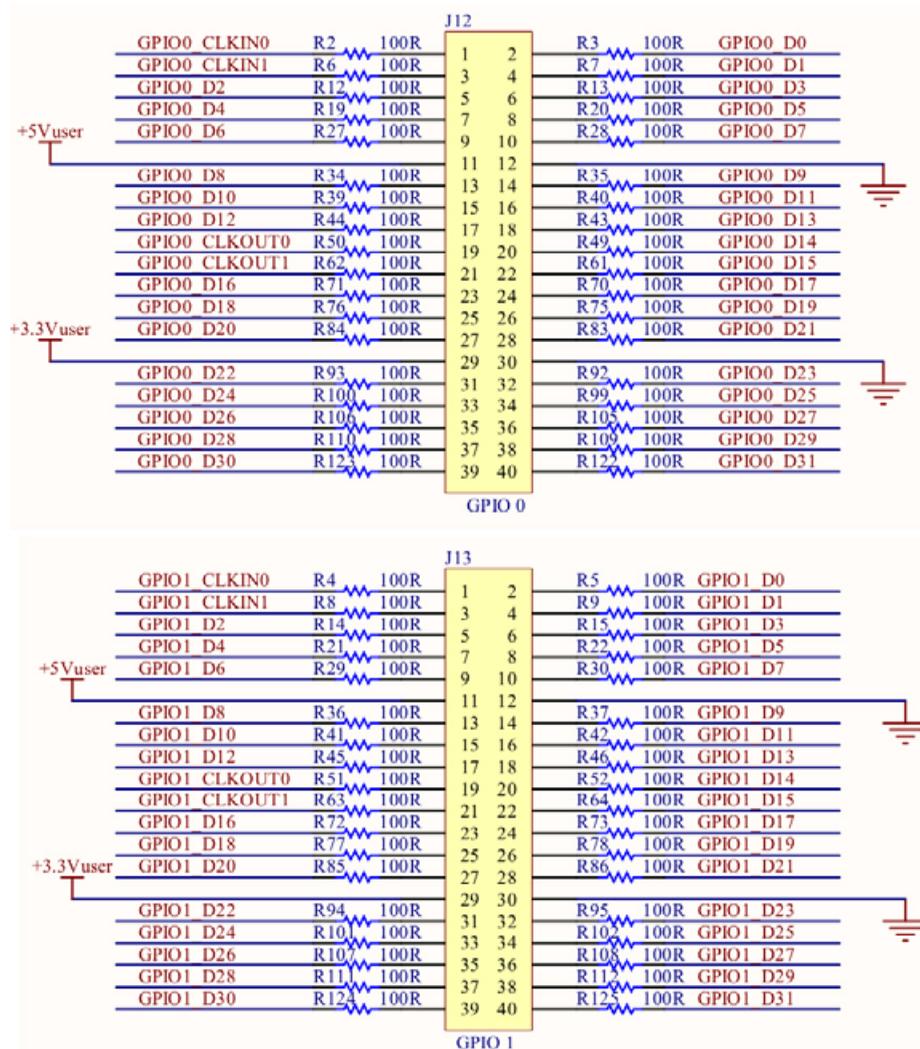


FIGURA 27 - CONECTORES GPIO 0 E GPIO 1 E A DESCRIÇÃO DE SEUS PINOS

TABELA 24 - PINAGEM DO FPGA RELACIONADA ÀS INTERFACES GPIO

NOME DO SINAL	PINO DO FPGA	PINO NO GPIO	DESCRIÇÃO
INTERFACE 0			
GPIO0_CLKIN[0]	G21	1	Entrada de clock 0 do GPIO0
GPIO0_D[0]	E16	2	Pino 0 de dado, entrada ou saída, do GPIO0
GPIO0_CLKIN[1]	G22	3	Entrada de clock 1 do GPIO0
GPIO0_D[1]	H22	4	Pino 1 de dado, entrada ou saída, do GPIO0
GPIO0_D[2]	F16	5	Pino 2 de dado, entrada ou saída, do GPIO0

GPIO0_D[3]	F19	6	Pino 3 de dado, entrada ou saída, do GPIO0
GPIO0_D[4]	H21	7	Pino 4 de dado, entrada ou saída, do GPIO0
GPIO0_D[5]	J22	8	Pino 5 de dado, entrada ou saída, do GPIO0
GPIO0_D[6]	F20	9	Pino 6 de dado, entrada ou saída, do GPIO0
GPIO0_D[7]	H19	10	Pino 7 de dado, entrada ou saída, do GPIO0
VCC	-	11	+5V
GND	-	12	GND
GPIO0_D[8]	J21	13	Pino 8 de dado, entrada ou saída, do GPIO0
GPIO0_D[9]	K21	14	Pino 9 de dado, entrada ou saída, do GPIO0
GPIO0_D[10]	H20	15	Pino 10 de dado, entrada ou saída, do GPIO0
GPIO0_D[11]	H18	16	Pino 11 de dado, entrada ou saída, do GPIO0
GPIO0_D[12]	L22	17	Pino 12 de dado, entrada ou saída, do GPIO0
GPIO0_D[13]	L21	18	Pino 13 de dado, entrada ou saída, do GPIO0
GPIO0_CLKOUT[0]	K18	19	Saída de clock 0 do GPIO0
GPIO0_D[14]	J18	20	Pino 14 de dado, entrada ou saída, do GPIO0
GPIO0_CLKOUT[1]	M22	21	Saída de clock 1 do GPIO0
GPIO0_D[15]	M21	22	Pino 15 de dado, entrada ou saída, do GPIO0
GPIO0_D[16]	K19	23	Pino 16 de dado, entrada ou saída, do GPIO0
GPIO0_D[17]	M19	24	Pino 17 de dado, entrada ou saída, do GPIO0
GPIO0_D[18]	N22	25	Pino 18 de dado, entrada ou saída, do GPIO0
GPIO0_D[19]	N21	26	Pino 19 de dado, entrada ou saída, do GPIO0
GPIO0_D[20]	M20	27	Pino 20 de dado, entrada ou saída, do GPIO0
GPIO0_D[21]	K17	28	Pino 21 de dado, entrada ou saída, do GPIO0
VCC	-	29	+3,3V
GND	-	30	GND
GPIO0_D[22]	N20	31	Pino 22 de dado, entrada ou saída, do GPIO0
GPIO0_D[23]	N19	32	Pino 23 de dado, entrada ou saída, do GPIO0
GPIO0_D[24]	P22	33	Pino 24 de dado, entrada ou saída, do GPIO0
GPIO0_D[25]	P21	34	Pino 25 de dado, entrada ou saída, do GPIO0

GPIO0_D[26]	N18	35	Pino 26 de dado, entrada ou saída, do GPIO0
GPIO0_D[27]	R19	36	Pino 27 de dado, entrada ou saída, do GPIO0
GPIO0_D[28]	R22	37	Pino 28 de dado, entrada ou saída, do GPIO0
GPIO0_D[29]	R21	38	Pino 29 de dado, entrada ou saída, do GPIO0
GPIO0_D[30]	R18	39	Pino 30 de dado, entrada ou saída, do GPIO0
GPIO0_D[31]	T18	40	Pino 31 de dado, entrada ou saída, do GPIO0
INTERFACE 1			
GPIO1_CLKIN[0]	A11	1	Entrada de clock 0 do GPIO1
GPIO1_D[0]	E11	2	Pino 0 de dado, entrada ou saída, do GPIO1
GPIO1_CLKIN[1]	B11	3	Entrada de clock 1 do GPIO1
GPIO1_D[1]	F11	4	Pino 1 de dado, entrada ou saída, do GPIO1
GPIO1_D[2]	E12	5	Pino 2 de dado, entrada ou saída, do GPIO1
GPIO1_D[3]	D13	6	Pino 3 de dado, entrada ou saída, do GPIO1
GPIO1_D[4]	E13	7	Pino 4 de dado, entrada ou saída, do GPIO1
GPIO1_D[5]	F13	8	Pino 5 de dado, entrada ou saída, do GPIO1
GPIO1_D[6]	G13	9	Pino 6 de dado, entrada ou saída, do GPIO1
GPIO1_D[7]	F14	10	Pino 7 de dado, entrada ou saída, do GPIO1
VCC	-	11	+5V
GND	-	12	GND
GPIO1_D[8]	E14	13	Pino 8 de dado, entrada ou saída, do GPIO1
GPIO1_D[9]	C15	14	Pino 9 de dado, entrada ou saída, do GPIO1
GPIO1_D[10]	D15	15	Pino 10 de dado, entrada ou saída, do GPIO1
GPIO1_D[11]	E15	16	Pino 11 de dado, entrada ou saída, do GPIO1
GPIO1_D[12]	C17	17	Pino 12 de dado, entrada ou saída, do GPIO1
GPIO1_D[13]	D17	18	Pino 13 de dado, entrada ou saída, do GPIO1
GPIO1_CLKOUT[0]	F15	19	Saída de clock 0 do GPIO1
GPIO1_D[14]	C19	20	Pino 14 de dado, entrada ou saída, do GPIO1
GPIO1_CLKOUT[1]	D19	21	Saída de clock 1 do GPIO1
GPIO1_D[15]	D20	22	Pino 15 de dado, entrada ou saída, do GPIO1

GPIO1_D[16]	G15	23	Pino 16 de dado, entrada ou saída, do GPIO1
GPIO1_D[17]	F17	24	Pino 17 de dado, entrada ou saída, do GPIO1
GPIO1_D[18]	G18	25	Pino 18 de dado, entrada ou saída, do GPIO1
GPIO1_D[19]	G17	26	Pino 19 de dado, entrada ou saída, do GPIO1
GPIO1_D[20]	H17	27	Pino 20 de dado, entrada ou saída, do GPIO1
GPIO1_D[21]	H16	28	Pino 21 de dado, entrada ou saída, do GPIO1
VCC	-	29	+3,3V
GND	-	30	GND
GPIO1_D[22]	G16	31	Pino 22 de dado, entrada ou saída, do GPIO1
GPIO1_D[23]	J17	32	Pino 23 de dado, entrada ou saída, do GPIO1
GPIO1_D[24]	M16	33	Pino 24 de dado, entrada ou saída, do GPIO1
GPIO1_D[25]	P17	34	Pino 25 de dado, entrada ou saída, do GPIO1
GPIO1_D[26]	P20	35	Pino 26 de dado, entrada ou saída, do GPIO1
GPIO1_D[27]	R17	36	Pino 27 de dado, entrada ou saída, do GPIO1
GPIO1_D[28]	R20	37	Pino 28 de dado, entrada ou saída, do GPIO1
GPIO1_D[29]	R16	38	Pino 29 de dado, entrada ou saída, do GPIO1
GPIO1_D[30]	T17	39	Pino 30 de dado, entrada ou saída, do GPIO1
GPIO1_D[31]	T16	40	Pino 31 de dado, entrada ou saída, do GPIO1

REFERÊNCIA 17 - INTERFACE GPIO

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
J12 e J13	Header, 20-Pin, Dual row	-	IDCSC-40

2.3.19 UTILIZANDO A PLACA EXPANSORA COM 16 CHAVES

Além das 4 chaves do tipo slide-switch presentes na placa Mercurio IV, é fornecida no kit a placa expansora com mais 16 chaves, dispostas em dois grupos de 8. O grupo da esquerda (posicionado bem em cima do conector GPIO) compreende as chaves do grupo A (SA[0..7]), e as da direita são as chaves do grupo B (SB[0..7]).

As chaves da placa expansora também estão ligadas a um filtro RC, porém, assim como na placa Mercurio IV, pode ser interessante utilizar processos de debounce para evitar glitches na entrada do FPGA.

A placa expansora pode ser conectada em uma das interfaces GPIO presentes na

placa Mercurio IV, portanto é importante observar os pinos utilizados pelas chaves da placa expansora para configurar a interface GPIO corretamente. É recomendado que a placa expansora seja conectada à interface GPIO0, por isso, a tabela de pinagem está relacionada com essa interface.

TABELA 25 - PINAGEM DA PLACA EXPANSORA

NOME DO SINAL	PINO DA GPIO0	PINO DO FPGA	DESCRIÇÃO
SA[0]	GPIO0_D[0]	E16	Chave 0 do grupo A
SA[1]	GPIO0_D[1]	H22	Chave 1 do grupo A
SA[2]	GPIO0_D[2]	F16	Chave 2 do grupo A
SA[3]	GPIO0_D[3]	F19	Chave 3 do grupo A
SA[4]	GPIO0_D[4]	H21	Chave 4 do grupo A
SA[5]	GPIO0_D[5]	J22	Chave 5 do grupo A
SA[6]	GPIO0_D[6]	F20	Chave 6 do grupo A
SA[7]	GPIO0_D[7]	H19	Chave 7 do grupo A
SB[0]	GPIO0_D[8]	J21	Chave 0 do grupo B
SB[1]	GPIO0_D[9]	K21	Chave 1 do grupo B
SB[2]	GPIO0_D[10]	H20	Chave 2 do grupo B
SB[3]	GPIO0_D[11]	H18	Chave 3 do grupo B
SB[4]	GPIO0_D[12]	L22	Chave 4 do grupo B
SB[5]	GPIO0_D[13]	L21	Chave 5 do grupo B
SB[6]	GPIO0_D[14]	J18	Chave 6 do grupo B
SB[7]	GPIO0_D[15]	M21	Chave 7 do grupo B

REFERÊNCIA 18 - CHAVES

REFERÊNCIA NA PLACA	DESCRIÇÃO	FABRICANTE	PART NUMBER
SA0 - SA8 SBO - SB8	Switch Slide SPDT Top Slide 0.3A 30VDC Gull Wing SMD T/R	-	JS102011SCQN

CAPÍTULO 3

3.1 - HISTÓRICO DE REVISÕES

VERSÃO	DESCRIÇÃO	AUTOR
V1	Manual do kit Mercurio IV 1.1b	Lucas Rotava lucas.rotava@macnicadhw.com.br
V2	Manual do kit Mercurio IV 2.0 (Troca dos componentes: conversores AD e DA, memória Flash, memória SDRAM e PHY Ethernet)	Lucas Rotava lucas.rotava@macnicadhw.com.br

3.2 - COPYRIGHT

Copyright © 2013 Macnica DHW Ltda. Todos os direitos reservados.



Rua Patrício Farias, 131,
Ed. Multicenter, Loja 01, CEP: 88034-132
Itacorubi, Florianópolis, SC, Brasil
Telefone: +55 (48) 3225.5052
Email: suporte@macnicadhw.com.br