



Implementação em FPGA de um conversor HDMI para transmissão em série de alta velocidade

Ana Marisa Oliveira Barbosa

VERSÃO DE TRABALHO

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Prof. Doutor João Paulo de Castro Canas Ferreira

Co-orientador: Prof. Dr. Henrique Manuel de Castro Faria Salgado

Supervisor Externo: Dr. Luís Manuel de Sousa Pessoa

1 de Junho de 2017

Conteúdo

1	HDMI	1
1.1	<i>Hardware</i> utilizado	1
1.1.1	Configurações da FPGA	2
1.1.2	Configuração dos <i>switches</i>	5
1.1.3	Arquiteturas Desenvolvidas	5
A	Descrição dos pinos das placas HDMI	15
B	Ficheiros da arquitetura A	19
B.0.1	Restrições Físicas	19
B.0.2	Restrições Temporais	21

Lista de Figuras

1.1	Ilustração dos sinais de som transmitidos no formato I^2S , retirada de [6]	4
1.2	Exemplo de imagem gerada pelo modulo desenvolvido	6
1.3	Máquina de estados para gerar uma barra de cores	7
1.4	Diagrama de blocos de arquitetura implementada utilizando um bloco gerador de barra de cores	9
1.5		12

Lista de Tabelas

1.1	Descrição e localização dos pinos de TB-FMCH-HDMI2 configurada por <i>default</i>	2
1.2	Descrição e localização dos pinos de TB-FMCH-HDMI2 configurada para 1 canal com suporte de áudio	3
1.3	Localização das portas de entrada e saída da arquitetura de transmissão de uma barra de cores para a placa HDMI transmissora	11
1.4	Localização das portas de entrada e saída da arquitetura de transmissão de uma imagem RGB de 10 bits entre as placas HDMI transmissora e receptora	14
A.1	Localização dos pinos de dados utilizados em TB-FMCH-HDMI2 configurado por <i>default</i>	16
A.2	Localização dos pinos de dados utilizados em TB-FMCH-HDMI2 com a configuração de um canal e suporte de áudio	17

Capítulo 1

HDMI

Este capítulo descreve o trabalho realizado para cumprir a primeira parte do projeto: obter uma conexão HDMI entre recetor e transmissor. São descritas as várias configurações das placas HDMI disponíveis e ainda as arquiteturas desenvolvidas e implementadas para cumprir esta parte do projeto.

1.1 *Hardware utilizado*

Tal como mencionado no sub-capítulo ??, para receber os dados provenientes do cabo HDMI e fazer a sua selecção são utilizadas duas placas HDMI (TB-FMCH-HDMI2 RX E TB-FMCH-HDMI2 TX) que, através das suas entradas e saída FMC de alta velocidade, conseguem enviar para a FPGA os sinais de imagem e som. Nas imagens ?? e ?? é possível visualizar o recetor (TB-FMCH-HDMI2 RX) e o transmissor (TB-FMCH-HDMI2 TX) HDMI utilizados neste projeto. Em conjunto, estas duas placas são designadas apenas por TB-FMCH-HDMI2. Estas mesmas placas são constituídas por conectores HDMI, de seguida o sinal é enviado para um recetor ou transmissor HDMI, ADV7612 no caso do recetor e ADV7511 no caso do transmissor. Finalmente os sinais provenientes do recetor/transmissor são enviados para uma FPGA embebida na placa (XC6SLX45-3FGG484C) que, consoante a sua configuração, envia pelos conectores FMC os sinais de áudio e vídeo.

As placas possuem ainda uma PROM (*Programmable read-only memory*) XCF16PFSG48C de configuração reprogramável que permite armazenar o *bitstream* que configura a FPGA embebida do modo que se pretende. É esta FPGA embebida que em cada placa (RX e TX) é responsável pela selecção e envio dos dados pretendidos para os conectores FMC (e posterior envio para a FPGA), e como tal é necessário que estejam configuradas para realizarem tais procedimentos. O recurso a estas memórias reconfiguráveis vem permitir uma fácil alteração da configuração da FPGA uma vez que, segundo [11], estas memórias de leitura permitem não só armazenar os *bitstreams* de configuração da FPGA, mas também reconfigurá-los, caso se pretenda, de uma forma fácil e eficiente.

1.1.1 Configurações da FPGA

A FPGA *Spartan-6* (XC6SLX45-3FGG484C) embebida nas placas tem 3 configurações disponíveis. Estas configurações variam não só no suporte que têm, que pode ser apenas de imagem mas também de imagem e áudio, mas variam também no número de bits por imagem que estas podem ter. Nas secções seguintes serão brevemente expostas as configurações disponíveis e como se pode tirar partido das mesmas no projeto.

→ Falar como é que se reconfigurou as placas

1.1.1.1 Configuração por *default*

Esta configuração vem previamente escrita na memória PROM de fábrica e acaba por ser a mais simples de todas. Os dados enviados pelos conectores FMC são apenas referentes aos dados de imagem. As tabelas ?? e ?? nas páginas ?? e ?? respectivamente identificam os pinos aos quais são atribuídas os sinais de dados de imagem HDMI tanto no recetor como no transmissor.

Esta configuração apenas transmite imagens RGB (*Red Green Blue*) com 10 bits. Assim sendo, tal como referido em [2], independentemente da formatação das imagens da fonte HDMI o recetor ADV7612 integrado na placa recetora HDMI converte a imagem para o formato RGB e transmite de maneira a enviar os dados em apenas 10 bits. FALAR SOBRE ESTAS CONVERSÕES

A tabela 1.1 da página 2, adaptada de [2], apresenta brevemente quais as portas das placas utilizadas e que sinais são transmitidos nas mesmas, no entanto é possível encontrar na tabela A.1 do anexo A mais pormenores relativamente a estes dados. Os nomes dos sinais são referentes aos sinais em TB-FMCH-HDMI2 (tanto TX como RX), assim como quando se faz referência à FPGA nestas tabelas estas correspondem às que estão embebidas nas placas HDMI.

PIN	FPGA -> FMC (RX)	FMC -> (TX)	Descrição
CLK0_M2C_P	RX#0_LLC	TX#0_DCLK	Sinal de relógio dos pixels
LA00_P_CC	RX#0_VSYNC	TX#0_VSYNC	Sincronização Vertical
LA01_P_CC	RX#0_HSYNC	TX#0_HSYNC	Sincronização Horizontal
LA02_P	RX#0_DE	TX#0_DE	Sinal de dados ativos
LA03_P a LA32_P	RX#0_P0 a RX#0_P29	TX#0_D0 a TX#0_D29	Pixel de Imagem

Tabela 1.1: Descrição e localização dos pinos de TB-FMCH-HDMI2 configurada por *default*

É de notar ainda que esta configuração é capaz de suportar até dois canais (RX0 e TX0, RX1 e TX1), no entanto nesta tabela apenas são apresentados os dados correspondentes ao canal 0 pois apenas será necessário utilizar um canal neste projeto.

Apesar de ser uma configuração simples, uma vez que apenas são transmitidos sinais de imagem em formato RGB, é uma configuração que será utilizada numa fase inicial em algumas arquiteturas implementadas que serão descritas no subcapítulo 1.1.3.

1.1.1.2 1 canal e suporte de audio

Para além da configuração descrita anteriormente em 1.1.1.1 que apenas suporta a transmissão de imagem, existe ainda uma configuração capaz de suportar não só a transmissão de imagem mas também de som. A configuração que é escrita na PROM para programar a FPGA embebida controla o recetor ADV7612 de maneira a conseguir transmitir imagens no formato YCbCr ou RGB com 12 bits e também fazer a transmissão do audio em formato I^2S .

Assim como referido em [12], neste caso a configuração da imagem está dependente da fonte HDMI, e é transmitida pelas placas tal como é emitida pela fonte, por outras palavras, se a fonte HDMI transmitir uma imagem em formato RGB é nesse formato que chega ao destino, no entanto se for transmitida uma imagem no formato YCbCr é nesse formato que chega ao seu destino. No caso do som, este é sempre transmitido em formato I^2S , o que implica a transmissão dos dados de audio mas também sinais de relógio necessários à sua transmissão. A tabela x passa a descrever com mais detalhe os sinais transmitidos para esta configuração da FPGA.

Na tabela 1.2 na página 3 são brevemente apresentados as portas e os sinais usados com este tipo de configuração da FPGA embebida. Na tabela A.2 no anexo A é apresentada uma tabela semelhante a esta, mas que inclui mais detalhes relativamente aos pinos usados e ao seu uso. Ambas as tabelas foram adaptadas de [12] onde é são apresentados todos os detalhes dos conectores FMC das placas.

PIN	FPGA ->FMC (RX)	FMC -> FPGA (TX)	FPGA->HDMI_TX
CLK0_M2C_P	RX#0_LLC	TX#0_DCLK	Sinal de relógio dos pixeis
LA00_P_CC	RX#0_VSYNC	TX#0_VSYNC	Sincronização vertical
LA01_P_CC	RX#0_HSYNC	TX#0_HSYNC	Sincronização horizontal
LA02_P	RX#0_DE	TX#0_DE	Sinal de dados ativos
LA03_P a LA32_P	RX#0_P0 a RX#0_P29	TX#0_D0 a TX#0_D29	Pixel de imagem do bit 0 ao 29
LA00_N_CC a LA01_N_CC	RX#0_InputVideoStatus	TX#0_InputVideoStatus	Formato de video (2D/3D)
LA19_N	RX#0_MCLK	TX#0_MCLK	Master Clock de som
LA20_N	RX#0_SCLK	TX#0_SCLK	Serial Clock de som
LA21_N a LA26_N	RX#0_AP0 a RX#0_AP5	TX#0_AP0 a TX#0_AP5	Dados de som
LA27_N a LA32_N	RX#0_P30 a RX#0_P35	TX#0_P30 a TX#0_P35	Pixel de imagem do bit 30 ao 35

Tabela 1.2: Descrição e localização dos pinos de TB-FMCH-HDMI2 configurada para 1 canal com suporte de audio

Os dados referentes ao som transmitidos pela placa recetor HDMI e recebidos de seguida pela placa emissora HDMI estão mencionados com mais detalhe na tabela A.2, e tal como indicado anteriormente, esta configuração é capaz de transmitir e receber dados no formato I^2S . Nas especificações deste protocolo, em [13], são definidos os sinais transmitidos aquando a utilização deste formato, que passam a ser descritos:

1. **Continuous Serial Clock (SCK):** Este sinal é por vezes reconhecido pelo nome de *Bit Clock* e é um sinal de relógio referente aos dados de som em série transmitidos pelos canais AP1, AP2, AP3 e AP4.
2. **Word Select(WS):** Este sinal é por vezes também conhecido por *Left/Right Clock* e é um sinal que indica o canal de som (esquerdo ou direito) que está a ser transmitido através dos dados em série recebidos ou enviados nas portas AP1, AP2, AP3 e AP4. É nomeado de sinal de relógio porque geralmente alterna entre 0 e 1 periodicamente, no entanto tal pode não acontecer, tal como referido em [13].
3. **Serial Data:** Sinais que transportam os dados de audio.

Na imagem 1.1 são ilustrados os sinais referentes ao audio descritos previamente. O sinal "SCLK"(Serial Clock) é referente ao sinal "Continuous Serial Clock", o sinal LRCLK (Left/Right Clock) refere-se ao sinal "Word Select" e ainda ISx refere-se ao sinal "Serial Data".

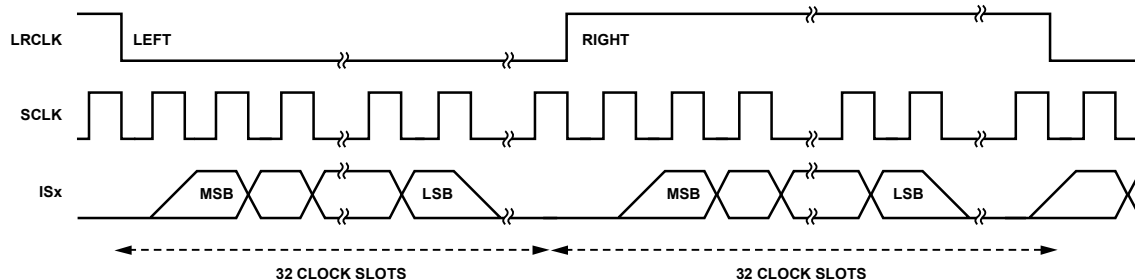


Figura 1.1: Ilustração dos sinais de som transmitidos no formato I^2S , retirada de [6]

Na placa recetora HDMI, que envia os dados para a FPGA Virtex-7, é também enviado o sinal *Master Clock* que corresponde a um sinal de relógio de referência dos sinais de áudio da entrada e ainda dados de áudio em AP0. É mencionado em [6] que estes dois sinais são referentes ao formato SPDIF e como tal não serão abordados neste projeto uma vez que as placas apenas suportam o formato I^2S .

Esta configuração das FPGAs embebidas nas placas é a configuração mais utilizada ao longo do projeto, uma vez que para além de ser versátil quanto ao formato das imagens transmitidas, é também capaz de suportar áudio. Em contrapartida, apenas suporta um canal (ao contrário da anterior), mas tal não é um problema pois apenas se pretende obter uma transmissão num único canal entre dispositivo de fonte HDMI e dispositivo final HDMI.

1.1.1.3 2 canais melhorado

→ Dizer que esta configuração faz tal tal e tal, mas que não será abordada uma vez que não foi utilizada

1.1.2 Configuração dos *switches*

Explicar como configurar os switches para obter o que queremos

1.1.3 Arquiteturas Desenvolvidas

Neste sub-capítulo passam a ser descritas as arquiteturas desenvolvidas e implementadas na FPGA referentes à comunicação entre as placas HDMI. Por outras palavras, é feita uma aplicação daquilo que foi explicado sobre as placas HDMI a serem utilizadas até agora em arquiteturas implementadas e testadas em FPGA.

1.1.3.1 Transmissão de uma imagem gerada na FPGA

Numa fase inicial do projeto, optou-se por simplificar a transmissão e para tal utilizou-se apenas a placa transmissora HDMI configurada por defeito. Construiu-se em Verilog um bloco capaz de gerar uma imagem para ser transmitida, mais especificamente uma barra de cores, e utilizou-se essa imagem para ser transmitida pelos conectores FMC.

O bloco gerador de uma barra de cores foi adaptado de um bloco disponibilizado pela *Inrenvium* aquando da compra das placas. Apesar de ter sido ligeiramente adaptado para este caso em específico, este baseia-se essencialmente numa máquina de estados que vai contando as linhas e as colunas para que possa enviar não só os valores das cores de cada pixel, mas também os sinais de controlo como a sincronização vertical, a sincronização horizontal e ainda os valores de pixels ativos.

Para que se entenda mais facilmente como e quando se transmitem os sinais de controlo da imagem e também os valores dos pixels é demonstrado na imagem 1.2 na página 6 um exemplo de transmissão de uma imagem gerada na FPGA. Antes de passar para descrição da geração da imagem passam a ser descritos os acrónimos apresentados na figura:

1. **HRES:** *Horizontal Resolution* é o parâmetro que define a resolução horizontal da imagem que vai ser gerada pelo bloco, ou seja o número de pixels em cada linha de transmissão.
2. **HSW:** *Horizontal Sync Width* é o parâmetro que define o número de ciclos de relógio que o sinal de sincronização horizontal tem.
3. **HBP:** *Horizontal Back Porch* é o parâmetro que define o número de pixels que não contêm informação útil (relativamente à cor dos mesmos) antes de começar a ser transmitida a linha de imagem.

4. **HFP:** *Horizontal Front Porch* é o parâmetro que define o número de pixels que não contêm informação útil depois de ser transmitida uma linha da imagem.
5. **VRES:** *Vertical Resolution* é o parâmetro que define a resolução vertical da imagem que vai ser gerada pelo bloco, por outras palavras é o número de linhas de pixels a ser geradas.
6. **VSW:** *Vertical Sync Width* é o parâmetro que define o número de linhas horizontais que o sinal de sincronização vertical está ativo.
7. **VBP:** *Vertical Back Porch* é o parâmetro que define o número de linhas horizontais que não contém informação útil relativamente ao pixels antes de começarem a ser transmitidas as linhas de pixels.
8. **VFP:** *Vertical Front Porch* é o parâmetro que define o número de linhas horizontais que não contém informação útil relativamente ao pixels depois de terem sido transmitidas todas as linhas horizontais da imagem.

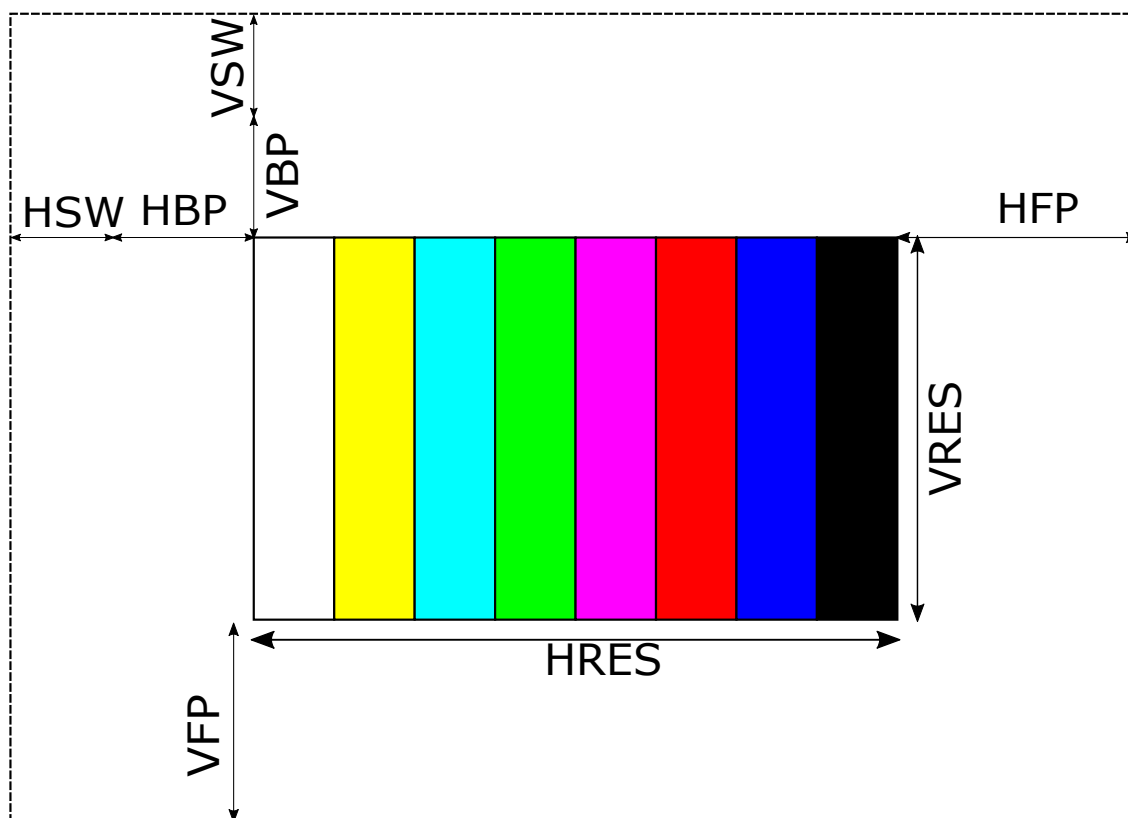


Figura 1.2: Exemplo de imagem gerada pelo modulo desenvolvido

Para gerar uma imagem em *FULL HD* cuja resolução é 1920x1080 pixels e o sinal de relógio deve ter uma frequência de 148.5 MHz, foram utilizados os seguintes valores para os parâmetros previamente descritos: HRES = 1920, HSW = 44, HBP = 44, HFP = 148, VRES = 1080, VSW = 5, VBP = 36 e VFP = 4.

A figura 1.3 na página 7 ilustra a máquina de estados desenvolvida para implementar a geração de uma barra a cores na FPGA.

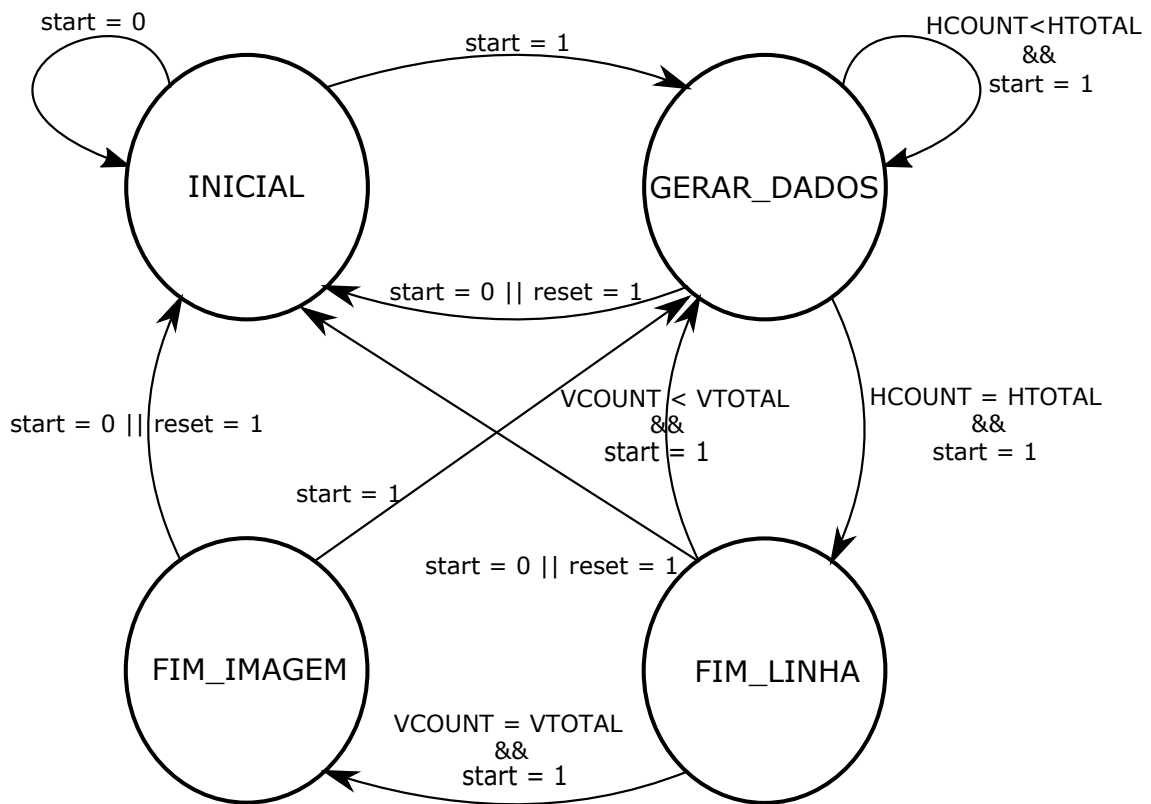


Figura 1.3: Máquina de estados para gerar uma barra de cores

Os registos VCOUNT E HCOUNT de decisão que se visualiza na figura correspondem a contadores que vão contando pixel a pixel até ao fim de uma linha (no caso do HCOUNT) ou então de uma imagem inteira (no caso do VCOUNT). Os valores de HTOTAL e VTOTAL não são mais do que a soma de todo o tamanho dos dados na horizontal e na vertical respectivamente. Assim sendo, para este caso em específico obtém-se os seguintes valores:

- $HTOTAL = HSW + HBP + HRES + HFP = 44 + 44 + 1920 + 148 = 2156$
- $VTOTAL = VSW + VBP + VRES + VFP = 5 + 36 + 1080 + 4 = 1125$

Para além destes sinais de decisão para mudança de estado existem mais dois sinais no diagrama da máquina de estados presente na figura 1.3 que ainda não foram mencionados que são o *reset* e o *start*. Estes dois sinais são botões do utilizador que lhe permitem definir quando se pretende que a transmissão esteja ativa ou desativa (através do botão *start*) ou então quando se pretende restabelecer os dados originais da máquina de estados (através do botão *reset*).

Existem 4 estados nesta máquina e consistem essencialmente em detecção do final de uma linha, e detecção do final de uma imagem e geração de dados. Os estados passam a ser descritos de seguida:

1. **Estado inicial:** Neste estado são configurados os parâmetros para o início de uma transmissão, ou seja, os valores de HCOUNT e VCOUNT são igualados ao valor total do tamanho na horizontal e na vertical respectivamente. Por outras palavras, os valores de HCOUNT e VCOUNT são igualados a HTOTAL e VTOTAL respectivamente. Isto acontece porque é possível retornar a este estado estando em qualquer um dos outros desde que seja pressionado o botão de *reset* ou então que a transmissão seja desligada pelo utilizador (*start* = 0).
2. **Estado para gerar dados:** Neste estado, ao flanco positivo do sinal de relógio do sistema, é incrementado o valor de HCOUNT e ao mesmo tempo são gerados os dados a serem transmitidos em cada ciclo de sinal de relógio, consoante o valor de HCOUNT e VCOUNT. Quando o valor de HCOUNT se igualar ao valor de HTOTAL, então significa que foi transmitida uma linha inteira da imagem, e por isso a máquina transita de estado e o valor de VCOUNT volta a ser igualado a 1. O processo de geração de dados será explicado em XXXX
3. **Estado de fim de linha:** Quando este estado está ativo, então uma linha da imagem foi transmitida, o que implica que é necessário incrementar o valor de linhas totais transmitidas (incrementando 1 valor em VCOUNT) e ainda verificar se a transmissão de uma imagem completa está realizada. Caso o valor de VCOUNT se iguale ao valor de VTOTAL, então transita-se para o estado de fim de imagem, e coloca-se o valor de VCOUNT a 1. Caso contrário, então a máquina transita para o estado que estava anteriormente.
4. **Estado de fim de imagem** Quando este estado está ativo então significa que ambos os valores de HCOUNT e VCOUNT estão igualados a 1 e que por isso já foi transmitida uma imagem completa e como tal passa-se a transmitir uma próxima imagem, transitando novamente para o estado para gerar dados.

Quando a máquina de estados se encontra no estado para gerar dados, então os dados de controlo são gerados nas seguintes condições :

- **Sinal de sincronização vertical:** O sinal de sincronização vertical é um sinal que como já foi referido anteriormente indica o início de transmissão de uma nova imagem, e por isso é ativado pela máquina de estados desenvolvida quando o valor em VCOUNT se igual ao valor de VTOTAL e quando o valor de HCOUNT se igual ao valor de HTOTAL, ou seja é ativado no final de uma imagem. Este sinal é ainda desligado quando o valor de VCOUNT se igual a VSW e o valor de HCOUNT se igual ao valor de HTOTAL, isto porque quando estas duas condições se verificam significa que o número de linhas em que o sinal de sincronização vertical deve estar ativo já terminou (é mesmo isso que o valor do parâmetro VSW define : *Vertical Sync Width*).
- **Sinal de sincronização horizontal:** O sinal de sincronização horizontal indica o início de uma nova linha e como tal deve ser ativo sempre que o valor de HCOUNT se igual e ao valor

de HTOTAL (porque indica o fim da emissão de uma linha). Da mesma maneira, este sinal deve ser desativo sempre que o valor de HCOUNT se igual ao valor de HSW, isto porque este valor indica que o período de tempo que este sinal deve estar ativo terminou.

- **Sinal de dados ativos:** Este sinal deve estar ativo sempre que se estiver a transmitir pixels válidos, e por isso sempre que as condições que serão de seguida apresentadas se verificarem:

1. O valor de VCCOUNT é maior do que a soma entre VSW e VBP.
2. O valor de VCOUNT é menor do que a soma entre VSW, VBP, VRES e 1.
3. O valor de HCOUNT é maior do que a soma entre HSW, HBP subtraída de 1 valor.
4. O valor de HCOUNT é menor do que a soma HSW, HBP e HRES.

As duas primeira condições garantem que VCOUNT está na zona vertical que corresponde à transmissão de imagem na figura 1.2, e as duas ultimas condições garantem o mesmo mas na zona horizontal.

- **Valor dos pixels:** Estes sinais correspondem a um barramento de 30 bits de uma imagem RGB com 10 bits por componente de cor. Como tal, estes valores devem corresponder a cores sempre o sinal de dados ativos estiver ligado e 0 sempre que estiver desligado.

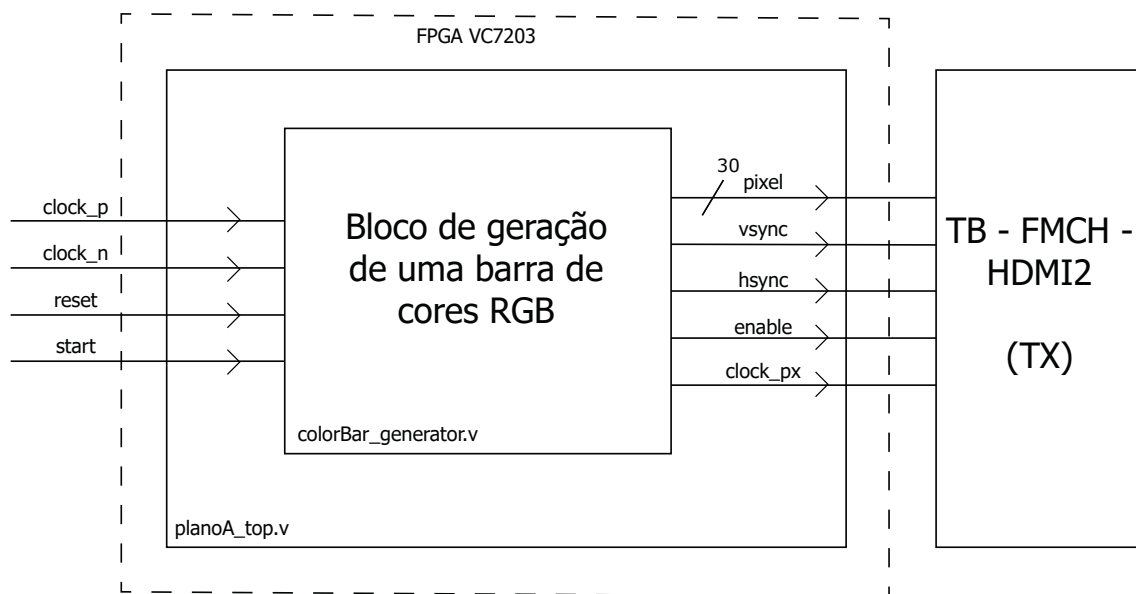


Figura 1.4: Diagrama de blocos de arquitetura implementada utilizando um bloco gerador de barra de cores

Na figura 1.4 é apresentado um diagrama de blocos da arquitetura implementada recorrendo a um bloco gerador de uma barra de cores. Este bloco foi implementado recorrendo-se à máquina de estados apresentada anteriormente.

Nas entradas do bloco estão ligados 4 sinais sendo que dois deles correspondem a um sinal de relógio diferencial de 200 MHz (*clock_p* corresponde ao sinal positivo e *clock_n* ao sinal negativo), e os outros dois sinais, *start* e *reset*, são sinais relevantes para a máquina de estados do bloco de geração de barras de cores definidos pelo utilizador e, por isso, são atribuídos a botões da FPGA. O sinal de relógio diferencial ligado às entradas deste bloco é proveniente do oscilador presente na FPGA e irá alimentar uma modulo que coloca na sua saída um sinal de relógio de 148.5 MHz. Esse módulo foi criado através do IP disponibilizado no VIVADO *Clocking Wizard* que vem facilitar a geração de um sinal de relógio com a frequência pretendida tendo como uma base um sinal diferencial de 200 MHz. O sinal gerado, de 148.5 MHz, é o sinal de relógio principal do sistema uma vez que é a frequência necessária para gerar uma imagem em *FULL HD*, e como tal é a essa cadência que os sinais serão enviados para a placa HDMI transmissora e é esse ainda o sinal de relógio da mesma.

Relativamente à saída do módulo é possível visualizar na imagem 1.4 que estas se encontram diretamente ligadas à placa transmissora HDMI através dos conectores FMC. Estes sinais são um barramento de 30 bits que corresponde ao pixel (*pixel*), o sinal de sincronização horizontal (*hsync*), o sinal de sincronização de vertical (*vsync*) e ainda o sinal de dados ativos (*enable*).

Para além do desenvolvimento do código em Verilog é necessário que as portas do modulo de topo, no caso desta arquitetura do modulo "planoA_top.v", estejam atribuídas a portas físicas da FPGA. Para tal é necessário definir onde estão as localizações das portas na FPGA (LOC) e o seu banco e criar um ficheiro de gere essas mesmas restrições físicas. A tabela 1.3 na página 11 indica quais as localizações físicas de cada porta existente no modulo de topo e no caso das saídas são também apresentados os nomes dos conectores na placa HDMI transmissora aos quais estas devem estar ligadas.

I/O	Sinal	LOC na FPGA	Banco na FPGA	Nome na placa HDMI	PIN da placa HDMI
O	clk_px	E34	35	TX#O_DCLK	CLK0_M2C_P
O	enable	K35	34	TX#0_DE	LA02_P
O	vsync	L31	34	TX#0_VSYNC	LA00_P_CC
O	hsync	M32	34	TX#0_HSYNC	LA01_P_CC
O	pixel[0]	J32	34	TX#0_D0	LA03_P
O	pixel[1]	K33	34	TX#0_D1	LA04_P
O	pixel[2]	L34	34	TX#0_D2	LA05_P
O	pixel[3]	M33	34	TX#0_D3	LA06_P
O	pixel[4]	H34	34	TX#0_D4	LA07_P
O	pixel[5]	K29	34	TX#0_D5	LA08_P
O	pixel[6]	J30	34	TX#0_D6	LA09_P
O	pixel[7]	L29	34	TX#0_D7	LA10_P
O	pixel[8]	J31	34	TX#0_D8	LA11_P
O	pixel[9]	M28	34	TX#0_D9	LA12_P
O	pixel[10]	R28	34	TX#0_D10	LA13_P

I/O	Sinal	LOC na FPGA	Banco na FPGA	Nome na placa HDMI	PIN da placa HDMI
O	pixel[11]	N28	34	TX#0_D11	LA14_P
O	pixel[12]	R30	34	TX#0_D12	LA15_P
O	pixel[13]	U31	34	TX#0_D13	LA16_P
O	pixel[14]	C35	35	TX#0_D14	LA17_P_CC
O	pixel[15]	D35	35	TX#0_D15	LA18_P_CC
O	pixel[16]	B36	35	TX#0_D16	LA19_P
O	pixel[17]	B34	35	TX#0_D17	LA20_P
O	pixel[18]	B39	35	TX#0_D18	LA21_P
O	pixel[19]	A35	35	TX#0_D19	LA22_P
O	pixel[20]	C38	35	TX#0_D20	LA23_P
O	pixel[21]	B37	35	TX#0_D21	LA24_P
O	pixel[22]	E32	35	TX#0_D22	LA25_P
O	pixel[23]	B32	35	TX#0_D23	LA26_P
O	pixel[24]	E33	35	TX#0_D24	LA27_P
O	pixel[25]	C33	35	TX#0_D25	LA28_P
O	pixel[26]	G32	35	TX#0_D26	LA29_P
O	pixel[27]	F36	35	TX#0_D27	LA30_P
O	pixel[28]	F34	35	TX#0_D28	LA31_P
O	pixel[29]	H33	35	TX#0_D29	LA32_P
I	clk_p	E19	38	—	—
I	clk_n	E18	38	—	—
I	reset	N41	19	—	—
I	start	E42	19	—	—

Tabela 1.3: Localização das portas de entrada e saída da arquitetura de transmissão de uma barra de cores para a placa HDMI transmissora

O ficheiro com estas restrições físicas gerado após a atribuição das mesmas é apresentado no sub-capítulo [B.0.1](#) do anexo [B](#). Para cada porta são atribuídas duas restrições: uma que indica a localização física na FPGA da porta e outra que indica a norma da mesma (*IOSTANDARD*). A primeira permite atribuir a um determinado lugar físico da FPGA a porta que se pretende e a segunda define a norma dessa mesma porta para que todas as considerações que se tenham de ser tomadas relativamente a essa porta tenham em conta essa mesma norma.

Para além destas restrições físicas geradas, são também geradas duas restrições temporais quanto aos sinais de relógio à entrada apresentadas no sub-capítulo [B.0.2](#) do anexo [B](#). As restrições temporais existentes definem que nas portas de entrada do sinal de relógio diferencial é mandatório haver um sinal com uma frequência de 200 MHz (período de 5ns). Isto porque este sinal de relógio é um sinal primário e como tal é importante que a ferramenta de síntese saiba o seu valor para poder garantir que toda a arquitetura cumpre os requisitos temporais.

Após a definição de todas as restrições e escrita do código em verilog, a arquitetura desenvolvida foi devidamente implementada na FPGA e testada obtendo-se o previsto.

1.1.3.2 Transmissão de imagem entre dispositivos HDMI

Na arquitetura desenvolvida que é apresentada neste sub-capítulo são utilizadas as placas HDMI receptora e transmissora ambas configuradas por defeito e procede-se à transmissão de uma imagem entre dispositivos HDMI. Foi desenvolvida uma arquitetura que recebe a cadência do sinal HDMI (neste caso em específico como é uma imagem *FULL HD* é uma frequência de 148,5 MHz) os sinais provenientes da placa HDMI receptora, mais especificamente o valor de *pixel*, *vsync*, *hsync* e *enable*.

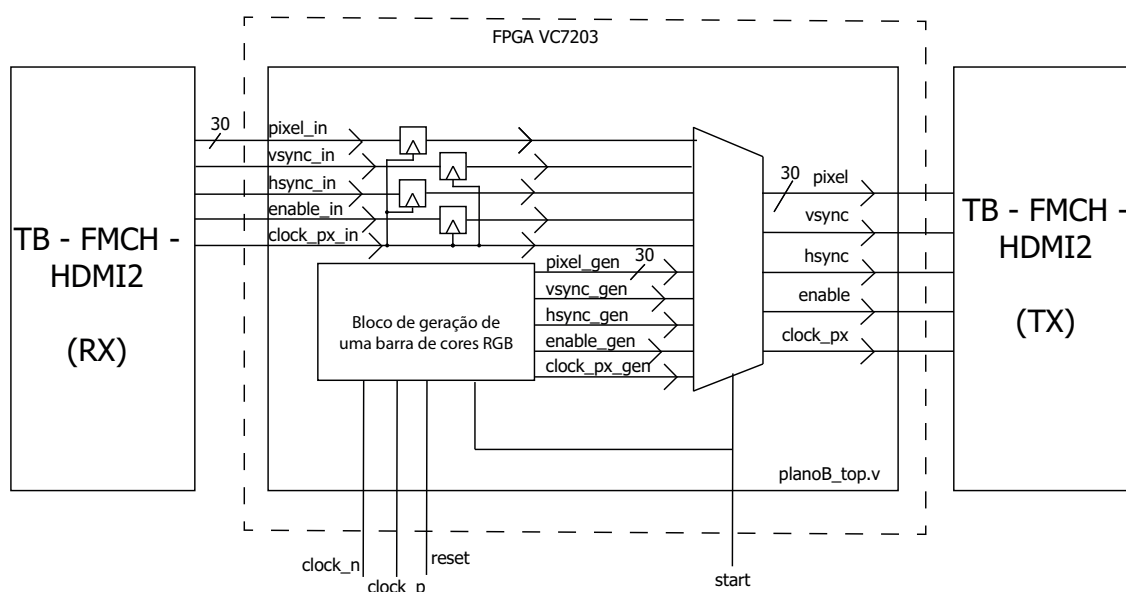


Figura 1.5

I/O	Sinal	LOC na FPGA	Banco na FPGA	Nome na placa HDMI	PIN da placa HDMI
I	clk_px_in	AJ32	14	RX#0_LLC	CLK0_M2C_P
I	enable_in	AN38	15	RX#0_VSYNC	LA02_P
I	vsync_in	AU38	15	RX#0_HSYNC	LA00_P_CC
I	hsync_in	AU39	15	RX#0_DE	LA01_P_CC
I	pixel_in[0]	AM41	15	RX#0_P0	LA03_P
I	pixel_in[1]	AR38	15	RX#0_P1	LA04_P
I	pixel_in[2]	AN40	15	RX#0_P2	LA05_P
I	pixel_in[3]	AR37	15	RX#0_P3	LA06_P
I	pixel_in[4]	AM39	15	RX#0_P4	LA07_P
I	pixel_in[5]	AP40	15	RX#0_P5	LA08_P

I/O	Sinal	LOC na FPGA	Banco na FPGA	Nome na placa HDMI	PIN da placa HDMI
I	pixel_in[6]	AP41	15	RX#0_P6	LA09_P
I	pixel_in[7]	AT39	15	RX#0_P7	LA10_P
I	pixel_in[8]	AR42	15	RX#0_P8	LA11_P
I	pixel_in[9]	AW37	15	RX#0_P9	LA12_P
I	pixel_in[10]	BA37	15	RX#0_P10	LA13_P
I	pixel_in[11]	AW38	15	RX#0_P11	LA14_P
I	pixel_in[12]	BB38	15	RX#0_P12	LA15_P
I	pixel_in[13]	BA39	15	RX#0_P13	LA16_P
I	pixel_in[14]	AK34	14	RX#0_P14	LA17_P_CC
I	pixel_in[15]	AJ33	14	RX#0_P15	LA18_P_CC
I	pixel_in[16]	AM36	14	RX#0_P16	LA19_P
I	pixel_in[17]	AJ36	14	RX#0_P17	LA20_P
I	pixel_in[18]	AP36	14	RX#0_P18	LA21_P
I	pixel_in[19]	AK37	14	RX#0_P19	LA22_P
I	pixel_in[20]	AN35	14	RX#0_P20	LA23_P
I	pixel_in[21]	AL36	14	RX#0_P21	LA24_P
I	pixel_in[22]	AG33	14	RX#0_P22	LA25_P
I	pixel_in[23]	AK35	14	RX#0_P23	LA26_P
I	pixel_in[24]	AH31	14	RX#0_P24	LA27_P
I	pixel_in[25]	AH34	14	RX#0_P25	LA28_P
I	pixel_in[26]	AM34	14	RX#0_P26	LA29_P
I	pixel_in[27]	AM31	14	RX#0_P27	LA30_P
I	pixel_in[28]	AM33	14	RX#0_P28	LA31_P
I	pixel_in[29]	AL29	14	RX#0_P29	LA32_P
O	clk_px	E34	35	TX#O_DCLK	CLK0_M2C_P
O	enable	K35	34	TX#0_DE	LA02_P
O	vsync	L31	34	TX#0_VSYNC	LA00_P_CC
O	hsync	M32	34	TX#0_HSYNC	LA01_P_CC
O	pixel[0]	J32	34	TX#0_D0	LA03_P
O	pixel[1]	K33	34	TX#0_D1	LA04_P
O	pixel[2]	L34	34	TX#0_D2	LA05_P
O	pixel[3]	M33	34	TX#0_D3	LA06_P
O	pixel[4]	H34	34	TX#0_D4	LA07_P
O	pixel[5]	K29	34	TX#0_D5	LA08_P
O	pixel[6]	J30	34	TX#0_D6	LA09_P
O	pixel[7]	L29	34	TX#0_D7	LA10_P
O	pixel[8]	J31	34	TX#0_D8	LA11_P

I/O	Sinal	LOC na FPGA	Banco na FPGA	Nome na placa HDMI	PIN da placa HDMI
O	pixel[9]	M28	34	TX#0_D9	LA12_P
O	pixel[10]	R28	34	TX#0_D10	LA13_P
O	pixel[11]	N28	34	TX#0_D11	LA14_P
O	pixel[12]	R30	34	TX#0_D12	LA15_P
O	pixel[13]	U31	34	TX#0_D13	LA16_P
O	pixel[14]	C35	35	TX#0_D14	LA17_P_CC
O	pixel[15]	D35	35	TX#0_D15	LA18_P_CC
O	pixel[16]	B36	35	TX#0_D16	LA19_P
O	pixel[17]	B34	35	TX#0_D17	LA20_P
O	pixel[18]	B39	35	TX#0_D18	LA21_P
O	pixel[19]	A35	35	TX#0_D19	LA22_P
O	pixel[20]	C38	35	TX#0_D20	LA23_P
O	pixel[21]	B37	35	TX#0_D21	LA24_P
O	pixel[22]	E32	35	TX#0_D22	LA25_P
O	pixel[23]	B32	35	TX#0_D23	LA26_P
O	pixel[24]	E33	35	TX#0_D24	LA27_P
O	pixel[25]	C33	35	TX#0_D25	LA28_P
O	pixel[26]	G32	35	TX#0_D26	LA29_P
O	pixel[27]	F36	35	TX#0_D27	LA30_P
O	pixel[28]	F34	35	TX#0_D28	LA31_P
O	pixel[29]	H33	35	TX#0_D29	LA32_P
I	reset	N41	19	—	—
I	start	E42	19	—	—
I	clk_p	E19	38	—	—
I	clk_n	E18	38	—	—

Tabela 1.4: Localização das portas de entrada e saída da arquitetura de transmissão de uma imagem RGB de 10 bits entre as placas HDMI transmissora e recetora

1.1.3.3 Transmissão de imagem e som entre dispositivos HDMI

Anexo A

Descrição dos pinos das placas HDMI

PIN	FPGA ->FMC (RX)	FMC->FPGA (TX)	Descrição
CLK0_M2C_P	RX#O_LLC	TX#O_DCLK	<i>clock</i> dos pixels
LA00_P_CC	RX#0_VSYNC	TX#0_VSYNC	sincr. Vertical
LA01_P_CC	RX#0_HSYNC	TX#0_HSYNC	sincr. Horizontal
LA02_P	RX#0_DE	TX#0_DE	sinal de dados ativos
LA03_P	RX#0_P0	TX#0_D0	Pixel de imagem B[0]
LA04_P	RX#0_P1	TX#0_D1	Pixel de imagem B[1]
LA05_P	RX#0_P2	TX#0_D2	Pixel de imagem B[2]
LA06_P	RX#0_P3	TX#0_D3	Pixel de imagem B[3]
LA07_P	RX#0_P4	TX#0_D4	Pixel de imagem B[4]
LA08_P	RX#0_P5	TX#0_D5	Pixel de imagem B[5]
LA09_P	RX#0_P6	TX#0_D6	Pixel de imagem B[6]
LA10_P	RX#0_P7	TX#0_D7	Pixel de imagem B[7]
LA11_P	RX#0_P8	TX#0_D8	Pixel de imagem B[8]
LA12_P	RX#0_P9	TX#0_D9	Pixel de imagem B[9]
LA13_P	RX#0_P10	TX#0_D10	Pixel de imagem G[0]
LA14_P	RX#0_P11	TX#0_D11	Pixel de imagem G[1]
LA15_P	RX#0_P12	TX#0_D12	Pixel de imagem G[2]
LA16_P	RX#0_P13	TX#0_D13	Pixel de imagem G[3]
LA17_P_CC	RX#0_P14	TX#0_D14	Pixel de imagem G[4]
LA18_P_CC	RX#0_P15	TX#0_D15	Pixel de imagem G[5]
LA19_P	RX#0_P16	TX#0_D16	Pixel de imagem G[6]
LA20_P	RX#0_P17	TX#0_D17	Pixel de imagem G[7]
LA21_P	RX#0_P18	TX#0_D18	Pixel de imagem G[8]
LA22_P	RX#0_P19	TX#0_D19	Pixel de imagem G[9]
LA23_P	RX#0_P20	TX#0_D20	Pixel de imagem R[0]
LA24_P	RX#0_P21	TX#0_D21	Pixel de imagem R[1]

PIN	FPGA ->FMC (RX)	FMC->FPGA (TX)	Descrição
LA25_P	RX#0_P22	TX#0_D22	Pixel de imagem R[2]
LA26_P	RX#0_P23	TX#0_D23	Pixel de imagem R[3]
LA27_P	RX#0_P24	TX#0_D24	Pixel de imagem R[4]
LA28_P	RX#0_P25	TX#0_D25	Pixel de imagem R[5]
LA29_P	RX#0_P26	TX#0_D26	Pixel de imagem R[6]
LA30_P	RX#0_P27	TX#0_D27	Pixel de imagem R[7]
LA31_P	RX#0_P28	TX#0_D28	Pixel de imagem R[8]
LA32_P	RX#0_P29	TX#0_D29	Pixel de imagem R[9]

Tabela A.1: Localização dos pinos de dados utilizados em TB-FMCH-HDMI2 configurado por *default*

PIN	FPGA-> (RX)	FMC -> FPGA (TX)	Descrição
CLK0_M2C_P	RX#O_LLC	TX#O_DCLK	Sinal de relógio dos pixels
LA00_P_CC	RX#0_VSYNC	TX#0_VSYNC	Sincronização vertical
LA01_P_CC	RX#0_HSYNC	TX#0_HSYNC	Sincronização horizontal
LA02_P	RX#0_DE	TX#0_DE	Sinal de dados ativos
LA03_P	RX#0_P0	TX#0_D0	Pixel de Imagem Cb[0]/B[0]
LA04_P	RX#0_P1	TX#0_D1	Pixel de Imagem Cb[1]/B[1]
LA05_P	RX#0_P2	TX#0_D2	Pixel de Imagem Cb[2]/B[2]
LA06_P	RX#0_P3	TX#0_D3	Pixel de Imagem Cb[3]/B[3]
LA07_P	RX#0_P4	TX#0_D4	Pixel de Imagem Cb[4]/B[4]
LA08_P	RX#0_P5	TX#0_D5	Pixel de Imagem Cb[5]/B[5]
LA09_P	RX#0_P6	TX#0_D6	Pixel de Imagem Cb[6]/B[6]
LA10_P	RX#0_P7	TX#0_D7	Pixel de Imagem Cb[7]/B[7]
LA11_P	RX#0_P8	TX#0_D8	Pixel de Imagem Cb[8]/B[8]
LA12_P	RX#0_P9	TX#0_D9	Pixel de Imagem Cb[9]/B[9]
LA13_P	RX#0_P10	TX#0_D10	Pixel de Imagem Cb[10]/B[10]
LA14_P	RX#0_P11	TX#0_D11	Pixel de Imagem Cb[11]/B[11]
LA15_P	RX#0_P12	TX#0_D12	Pixel de Imagem Y[0]/B[0]
LA16_P	RX#0_P13	TX#0_D13	Pixel de Imagem Y[1]/B[1]
LA17_P_CC	RX#0_P14	TX#0_D14	Pixel de Imagem Y[2]/B[2]
LA18_P_CC	RX#0_P15	TX#0_D15	Pixel de Imagem Y[3]/B[3]
LA19_P	RX#0_P16	TX#0_D16	Pixel de Imagem Y[4]/B[4]
LA20_P	RX#0_P17	TX#0_D17	Pixel de Imagem Y[5]/B[5]
LA21_P	RX#0_P18	TX#0_D18	Pixel de Imagem Y[6]/B[6]
LA22_P	RX#0_P19	TX#0_D19	Pixel de Imagem Y[7]/B[7]
LA23_P	RX#0_P20	TX#0_D20	Pixel de Imagem Y[8]/B[8]

PIN	FPGA-> (RX)	FMC -> FPGA (TX)	Descrição
LA24_P	RX#0_P21	TX#0_D21	Pixel de Imagem Y[9]/B[9]
LA25_P	RX#0_P22	TX#0_D22	Pixel de Imagem Y[10]/B[10]
LA26_P	RX#0_P23	TX#0_D23	Pixel de Imagem Y[11]/B[11]
LA27_P	RX#0_P24	TX#0_D24	Pixel de Imagem Cr[0]/R[0]
LA28_P	RX#0_P25	TX#0_D25	Pixel de Imagem Cr[1]/R[1]
LA29_P	RX#0_P26	TX#0_D26	Pixel de Imagem Cr[2]/R[2]
LA30_P	RX#0_P27	TX#0_D27	Pixel de Imagem Cr[3]/R[3]
LA31_P	RX#0_P28	TX#0_D28	Pixel de Imagem Cr[4]/R[4]
LA32_P	RX#0_P29	TX#0_D29	Pixel de Imagem Cr[5]/R[5]
LA00_N_CC	RX#0_InputVideoStatus[0]	TX#0_InputVideoStatus[0]	Formato do video (2D/3D)
LA01_N_CC	RX#0_InputVideoStatus[1]	TX#0_InputVideoStatus[1]	Formato do video (2D/3D)
LA19_N	RX#0_MCLK	TX#0_MCLK	<i>Master Clock</i> de som
LA20_N	RX#0_SCLK	TX#0_SCLK	<i>Serial Clock</i> de som
LA21_N	RX#0_AP0	TX#0_AP0	Dados de Som SPDIF
LA22_N	RX#0_AP1	TX#0_AP1	Dados de Som I2S [0]
LA23_N	RX#0_AP2	TX#0_AP2	Dados de Som I2S [1]
LA24_N	RX#0_AP3	TX#0_AP3	Dados de Som I2S [2]
LA25_N	RX#0_AP4	TX#0_AP4	Dados de Som I2S [3]
LA26_N	RX#0_AP5	TX#0_AP5	Sinal de relógio LR (left/right)
LA27_N	RX#0_P30	TX#0_D30	Pixel de Imagem Cr[6]/R[6]
LA28_N	RX#0_P31	TX#0_D31	Pixel de Imagem Cr[7]/R[7]
LA29_N	RX#0_P32	TX#0_D32	Pixel de Imagem Cr[8]/R[8]
LA30_N	RX#0_P33	TX#0_D33	Pixel de Imagem Cr[9]/R[9]
LA31_N	RX#0_P34	TX#0_D34	Pixel de Imagem Cr[10]/R[10]
LA32_N	RX#0_P35	TX#0_D35	Pixel de Imagem Cr[11]/R[11]

Tabela A.2: Localização dos pinos de dados utilizados em TB-FMCH-HDMI2 com a configuração de um canal e suporte de audio

Anexo B

Ficheiros da arquitetura A

B.0.1 Restrições Físicas

```
set_property PACKAGE_PIN E18 [get_ports clk_n]
set_property IOSTANDARD DIFF_HSTL_II_18 [get_ports clk_p]
set_property PACKAGE_PIN E34 [get_ports clk_out]
set_property PACKAGE_PIN K35 [get_ports enable]
set_property IOSTANDARD LVCMOS18 [get_ports clk_out]
set_property IOSTANDARD LVCMOS18 [get_ports enable]
set_property IOSTANDARD LVCMOS18 [get_ports vsync]
set_property IOSTANDARD LVCMOS18 [get_ports hsync]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[29]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[28]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[27]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[26]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[25]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[24]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[23]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[22]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[21]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[20]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[19]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[18]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[17]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[16]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[15]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[14]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[13]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[12]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[11]]
```

```
set_property IOSTANDARD LVCMOS18 [get_ports pixel[10]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[9]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[8]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[7]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[6]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[5]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[4]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[3]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[2]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[1]]
set_property IOSTANDARD LVCMOS18 [get_ports pixel[0]]
set_property IOSTANDARD LVCMOS18 [get_ports reset]
set_property IOSTANDARD LVCMOS18 [get_ports start]
set_property PACKAGE_PIN N41 [get_ports reset]
set_property PACKAGE_PIN M32 [get_ports hsync]
set_property PACKAGE_PIN L31 [get_ports vsync]
set_property PACKAGE_PIN J32 [get_ports pixel[0]]
set_property PACKAGE_PIN K33 [get_ports pixel[1]]
set_property PACKAGE_PIN L34 [get_ports pixel[2]]
set_property PACKAGE_PIN M33 [get_ports pixel[3]]
set_property PACKAGE_PIN H34 [get_ports pixel[4]]
set_property PACKAGE_PIN K29 [get_ports pixel[5]]
set_property PACKAGE_PIN J30 [get_ports pixel[6]]
set_property PACKAGE_PIN L29 [get_ports pixel[7]]
set_property PACKAGE_PIN J31 [get_ports pixel[8]]
set_property PACKAGE_PIN M28 [get_ports pixel[9]]
set_property PACKAGE_PIN R28 [get_ports pixel[10]]
set_property PACKAGE_PIN N28 [get_ports pixel[11]]
set_property PACKAGE_PIN R30 [get_ports pixel[12]]
set_property PACKAGE_PIN U31 [get_ports pixel[13]]
set_property PACKAGE_PIN C35 [get_ports pixel[14]]
set_property PACKAGE_PIN D35 [get_ports pixel[15]]
set_property PACKAGE_PIN B36 [get_ports pixel[16]]
set_property PACKAGE_PIN B34 [get_ports pixel[17]]
set_property PACKAGE_PIN B39 [get_ports pixel[18]]
set_property PACKAGE_PIN A35 [get_ports pixel[19]]
set_property PACKAGE_PIN C38 [get_ports pixel[20]]
set_property PACKAGE_PIN B37 [get_ports pixel[21]]
set_property PACKAGE_PIN E32 [get_ports pixel[22]]
set_property PACKAGE_PIN B32 [get_ports pixel[23]]
```

```
set_property PACKAGE_PIN E33 [get_ports pixel[24]]  
set_property PACKAGE_PIN C33 [get_ports pixel[25]]  
set_property PACKAGE_PIN G32 [get_ports pixel[26]]  
set_property PACKAGE_PIN F36 [get_ports pixel[27]]  
set_property PACKAGE_PIN F34 [get_ports pixel[28]]  
set_property PACKAGE_PIN H33 [get_ports pixel[29]]  
set_property PACKAGE_PIN E42 [get_ports start]
```

B.0.2 Restrições Temporais

```
create_clock -period 5.000 [get_ports clk_n]  
create_clock -period 5.000 [get_ports clk_p]
```


Bibliografia

- [1] Xilinx and Inc, “VC7203 Virtex-7 FPGA GTX Transceiver Characterization Board User Guide (UG957),” 2014.
- [2] Inrevium, *Manual do Utilizador de TB-FMCH-HDMI2 Hardware*. 2012.
- [3] D. Chen, “SerDes Transceivers for High-speed Serial Communications,”
- [4] Xilinx and Inc, “Xilinx WP431 Leveraging 7 Series FPGA Transceivers for High-Speed Serial I/O Connectivity, White Paper,” 2013.
- [5] Xilinx and Inc, “7 Series FPGAs GTX/GTH Transceivers User Guide (UG476),”
- [6] Analog Devices, “ADV7612 Reference Manual,”
- [7] Wikipedia Contributors, “HDMI,” 2016.
- [8] S. Koenig, D. Lopez-Diaz, J. Antes, F. Boes, R. Henneberger, A. Leuther, A. Tessmann, R. Schmogrow, D. Hillerkuss, R. Palmer, T. Zwick, C. Koos, W. Freude, O. Ambacher, J. Leuthold, and I. Kallfass, “Wireless sub-THz communication system with high data rate enabled by RF photonics and active MMIC technology,” *2014 IEEE Photonics Conference, IPC 2014*, vol. 7, no. December 2013, pp. 414–415, 2014.
- [9] J. Federici and L. Moeller, “Review of terahertz and subterahertz wireless communications,” *Journal of Applied Physics*, vol. 107, no. 11, 2010.
- [10] W. contributors, “audio and video interfaces and connectors,” 2016.
- [11] Xilinx, “Platform Flash In-System,” *Memory*, vol. 123, pp. 1–46, 2006.
- [12] Inrevium, “TB-FMCH-HDMI2 Hardware User Manual 1 IN / OUT + Audio support,” pp. 1–59.
- [13] P. Semiconductors and B. I. Timing, “I²S bus specification I²S bus specification,” no. February 1986, pp. 1–7, 1996.