Traduzido do Inglês para o Português - www.onlinedoctranslator.com

Interfaces principais MicroBlaze

Criação da Plataforma de Hardware

Exemplos de núcleos IP

Desenvolvimento de software

AULA 6

IOUL IIA SKL I AROVA

Universidade de Aveiro

Interfaces MicroBlaze

MicroBlaze não separa acessos de dados para E/S e memória (ele usa E/S mapeada em memória).

O processador possui até três interfaces para acesso à memória:

- Barramento de memória local (LMB) que fornece acesso de ciclo único ao bloco de RAM de porta dupla no chip.
- Advanced eXtensible Interface (AXI4) para conexão com periféricos e memória on-chip e offchip.
- Advanced eXtensible Interface (AXI4) ou AXI Coherency Extension (ACE) para conexões coerentes de cache com a memória.

MicroBlaze também suporta até 16 portas de interface AXI4-Stream, cada uma com uma interface mestre e uma interface escrava.

As interfaces no MicroBlaze têm 32 bits de largura.

Interfaces principais MicroBlaze

M_AXI_DP: Interface de dados periféricos, interface AXI4-Lite ou AXI4

DLMB: Interface de dados, barramento de memória local (somente BRAM)

M_AXI_IP: Interface de instrução periférica, interface AXI4-Lite

ILMB: Interface de instrução, barramento de memória local (somente BRAM)

M0_AXIS..M15_AXIS: Interfaces de conexão direta mestre da interface AXI4-Stream

S0_AXIS..S15_AXIS: Interfaces de conexão direta escrava da interface AXI4-Stream

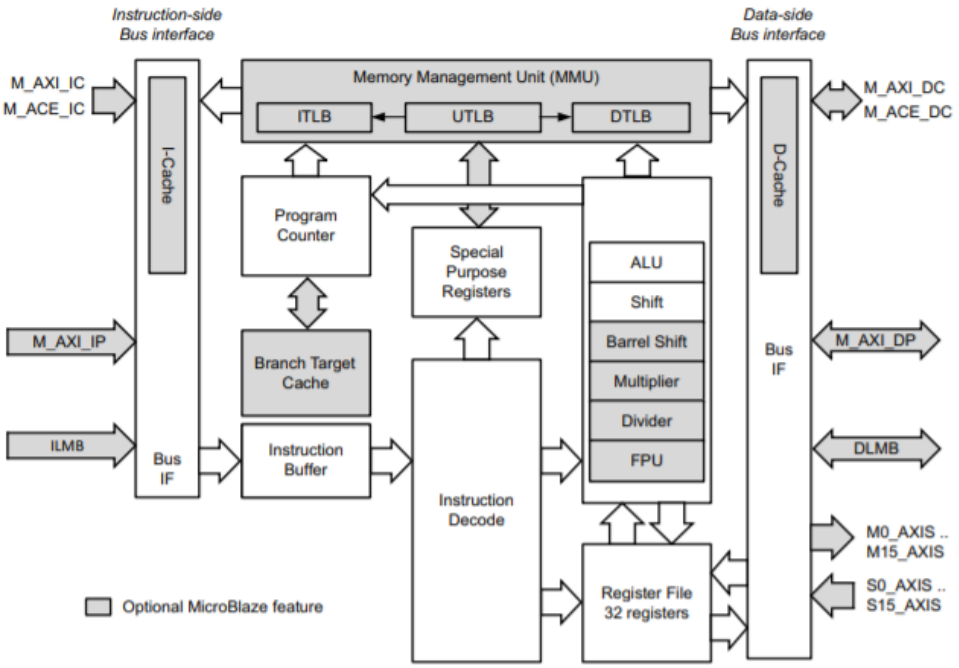
M_AXI_DC: Interface AXI4 do cache do lado dos dados

M_ACE_DC: Interface AXI Coherency Extension (ACE) do cache do lado dos dados

M_AXI_IC: Interface AXI4 do cache do lado da instrução

M_ACE_IC: Interface AXI Coherency Extension (ACE) do cache do lado da instrução

Essencial: Sinais diversos para: clock, reset, interrupção, depuração, rastreamento



EIXO

Interface extensível avançadaé uma interconexão ponto a ponto projetada para sistemas microcontroladores de alto desempenho e alta velocidade.

O**EIXO**O protocolo é baseado em uma interconexão ponto a ponto para evitar o compartilhamento de barramento e, portanto, permitir maior largura de banda e menor latência.

Existem três tipos de interfaces AXI4:

- AXI4-Lite — para comunicação mapeada em memória simples e de baixo rendimento, fornecendo uma estrutura semelhante a um registro com recursos e complexidade reduzidos (1 transferência por transação)
- AXI4 — para requisitos de mapeamento de memória de alto desempenho (até 256 transferências de dados)
- AXI4-Stream — para streaming de dados em alta velocidade (quantidade ilimitada de dados)

As especificações AXI descrevem uma interface entre um único mestre AXI e um único escravo AXI.

Interconexão AXIpermitir que vários mestres e/ou vários escravos interajam entre si.

Na realidade, as interconexões contêm interfaces escravas que se conectam aos mestres AXI e interfaces mestres que se conectam aos escravos AXI. O que acontece em uma interconexão – ou seja, como diferentes mestres se comunicam com diferentes escravos – depende da implementação. As interconexões podem permitir um barramento de endereço compartilhado, um barramento de dados compartilhado, ambos compartilhados ou nenhum compartilhado.

Canais AXI4 e AXI4-Lite

Existem cinco canais independentes entre um mestre e um escravo AXI.

Os canais de endereço são usados para enviar informações de endereço e controle enquanto executa um handshake básico entre mestre e escravo.

Um mestre lê e grava dados em um escravo. As informações de resposta de leitura são colocadas no canal de dados de leitura, enquanto a resposta de gravação informações tem um canal dedicado. Desta forma, o mestre pode verificar se uma transação de gravação foi concluída.

Toda troca de dados é chamada de transação. Uma transação inclui o endereço e as informações de controle, os dados enviados, bem como qualquer informação de resposta. Os dados reais são enviados em rajadas que contêm múltiplas transferências.

Visão geral da depuração

MicroBlaze apresenta uma interface de depuração para suportar ferramentas de depuração de software baseadas em JTAG.

A interface de depuração foi projetada para ser conectada ao núcleo do Módulo de depuração do microprocessador Xilinx (MDM), que faz interface com a porta JTAG dos FPGAs Xilinx.

Para poder baixar programas, definir pontos de interrupção de software e desmontar código, os intervalos de memória de instruções e de dados devem se sobrepor e usar a mesma memória física.

Universidade de Aveiro

Exemplo de projeto baseado em MicroBlaze

Processador MicroBlaze

Instrução local e memória de dados

Unidades de relógio e reset

Portas de E/S de uso geral (conectadas a LEDs, interruptores, botões e displays de 7 segmentos Nexys-4)

UART RS232

Interconexão AXI (barra transversal para interconexão de microprocessador e periféricos)

Controlador de interrupção

Universidade de Aveiro

Instalação do arquivo da placa

Configure a placa:

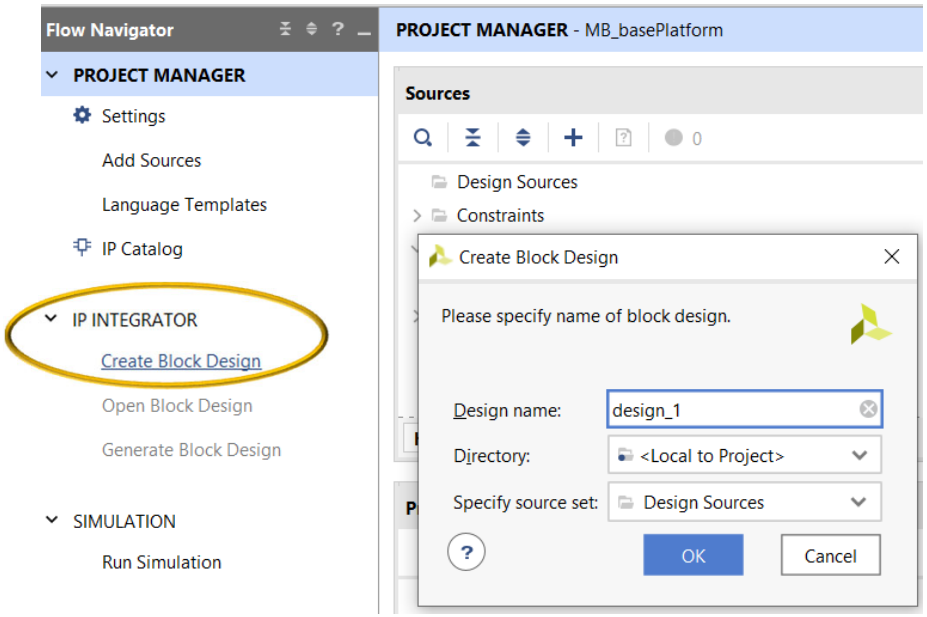
- Não use o recurso Instalar/Atualizar ao criar um novo projeto
- Baixe os arquivos do tabuleiro em**Ficheiro "Nexys4.rar" (arquivos de suporte da placa Nexys-4 para Xilinx Vivado)** no eLearning
- Copie a pasta nexys4 para a pasta de instalação do Vivado (C:\Xilinx\Vivado\2022.2\data\boards\board_files)
  - Por padrão, esta pasta contém arquivos XML para diferentes placas FPGA fabricadas pela Xilinx
  - Os arquivos XML definem várias interfaces na placa, como interruptores deslizantes, botões, LEDs, USB-UART, memória, Ethernet etc.
- Reinicie o Vivado
- Agora você está pronto para iniciar um novo projeto Vivado baseado em IP Integrator para a placa Nexys-4
- Se não estiver funcionando, execute TCL:
  - set_param placa.repoPaths ""
  - set_param board.repoPaths {C:/Xilinx/Vivado/2022.2/data/boards/board_files/nexys4}

Integrador IP

À medida que os FPGAs se tornam maiores e mais complexos, e à medida que os cronogramas de projeto se tornam mais curtos, o uso de IP de terceiros e a reutilização de projetos se tornam obrigatórios.

Integrador IPtem como objetivo ajudar os designers com questões de design e reutilização de IP.

O integrador Vivado IP permite criar projetos de sistemas complexos instanciando e interconectando IP do catálogo Vivado IP em uma tela de design.

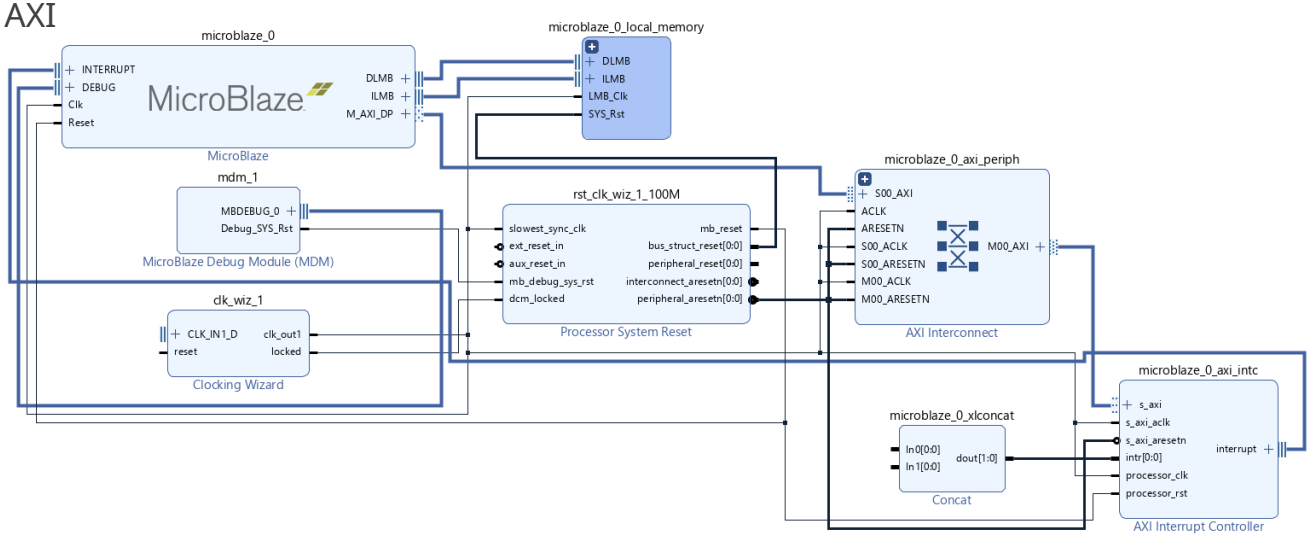


Automação de blocos

Adicionar IP MicroBlaze

Automação de blocos ajuda você a montar um sistema MicroBlaze básico que consiste no seguinte:

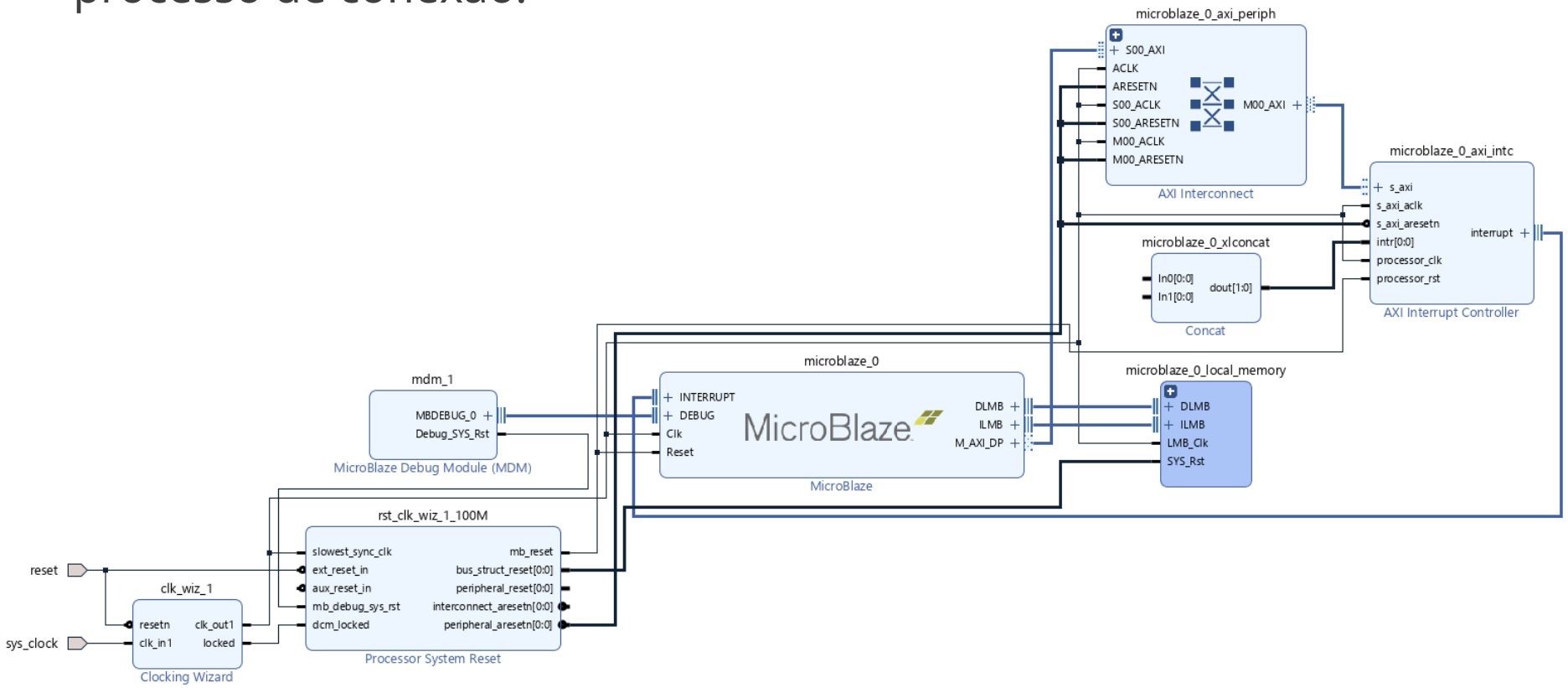
- Um módulo de depuração MicroBlaze
- Um bloco hierárquico chamado microblaze_0_local_memory
- Um assistente de relógio
- Um módulo de reinicialização
- Uma interconexão AXI
- Um controlador de interrupção AXI



Automação de conexão

Automação de conexão auxilia você a fazer conexões internas entre diferentes blocos e conexões com interfaces externas.

Vários ajustes devem ser feitos para orientar e realizar o processo de conexão.



Universidade de Aveiro

Interface externa

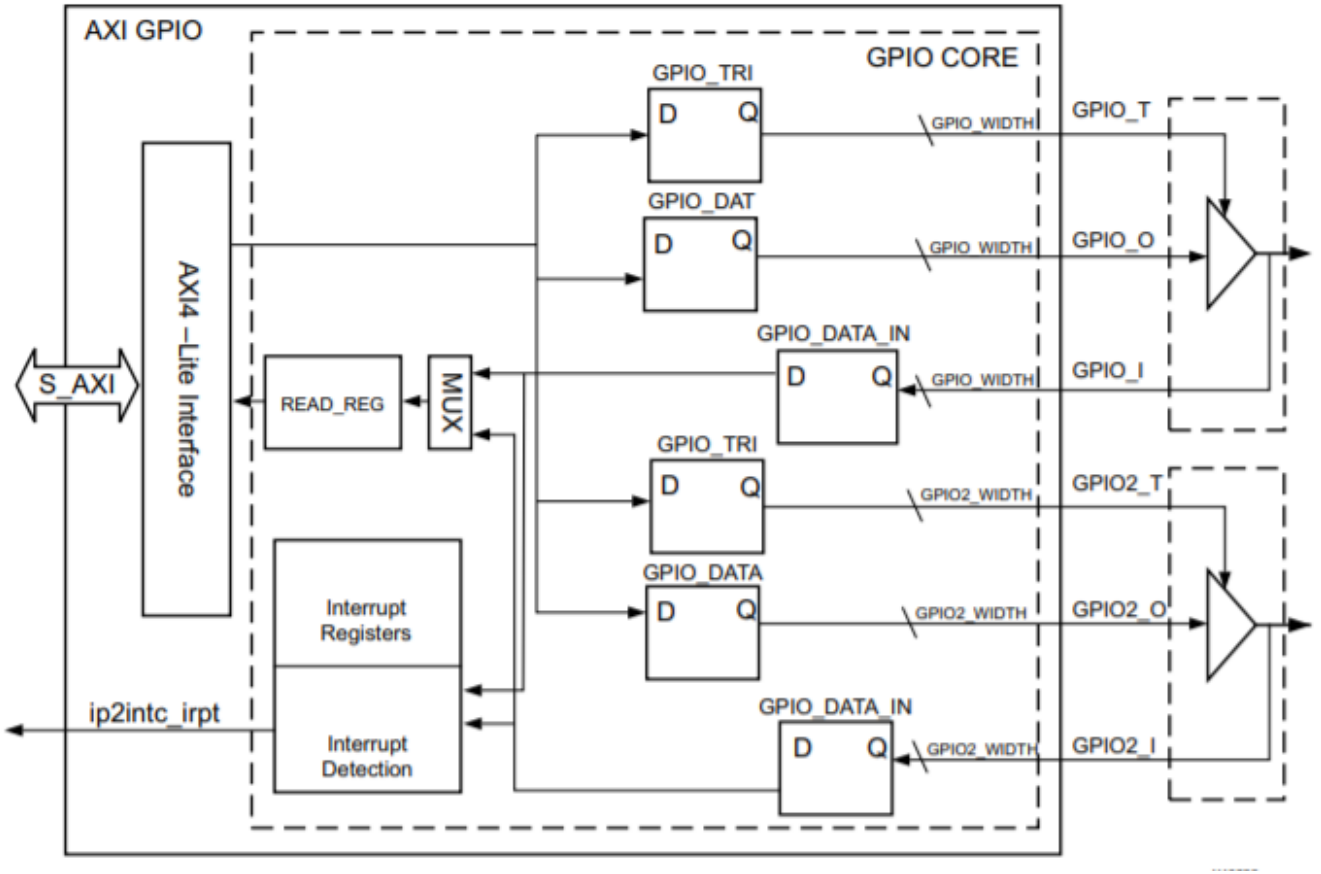
GPIOA interface pode ser vinculada a uma das diversas interfaces presentes na placa de destino.

Núcleo GPIOfornece uma interface de entrada/saída de uso geral para a interface AXI. Este núcleo de soft IP de 32 bits suporta:

- a especificação da interface AXI4-Lite
- canais GPIO simples ou duplos configuráveis
- largura de canal configurável para pinos GPIO de 1 a 32 bits
- programação dinâmica de cada bit GPIO como entrada ou saída
- configuração individual de cada canal
- valores de reinicialização independentes para cada bit de todos os registros
- geração de solicitação de interrupção opcional

AXIGPIO

O núcleo GPIO consiste em registradores e multiplexadores para leitura e gravação dos registradores do canal AXI GPIO. Também inclui a lógica necessária para identificar um evento de interrupção quando a entrada do canal muda.



Registros GPIO

O registro de dados AXI GPIO é usado para ler as portas de entrada de uso geral e gravar nas portas de saída de uso geral. Quando uma porta é configurada como entrada, a gravação no registro de dados AXI GPIO não tem efeito.

O registro de controle de 3 estados AXI GPIO é usado para configurar as portas dinamicamente como entrada ou saída. Quando umpedaçodentro deste registro édefinir, a porta de E/S correspondente é configurada comoporta de entrada. Quando umpedaçoé limpo, a porta de E/S correspondente é configurada comoporta de saída.

Address Space Offset(3)	Register Name	Access Type	Default Value	Description
0x0000	GPIO_DATA	R/W	0x0	Channel 1 AXI GPIO Data Register.
0x0004	GPIO_TRI	R/W	0x0	Channel 1 AXI GPIO 3-state Control Register.
0x0008	GPIO2_DATA	R/W	0x0	Channel 2 AXI GPIO Data Register.
0x000C	GPIO2_TRI	R/W	0x0	Channel 2 AXI GPIO 3-state Control.
0x011C	GIER(1)	R/W	0x0	Global Interrupt Enable Register.
0x0128	IP IER(1)	R/W	0x0	IP Interrupt Enable Register (IP IER).
0x0120	IP ISR(1)	R/TOW(2)	0x0	IP Interrupt Status Register.

Configuração GPIO

Para portas de entrada quando a interrupção não está habilitada, siga as seguintes etapas:

- Configure a porta como entrada escrevendo o bit correspondente no registro GPIOx_TRI com o valor 1.
- Leia o bit correspondente no registro GPIOx_DATA.

Para portas de saída, use as seguintes etapas:

- Configure a porta como saída escrevendo o bit correspondente no registro GPIOx_TRI com valor 0.
- Escreva o bit correspondente no registro GPIOx_DATA.

Editor de endereços

O editor de endereços é uma visualização de tabela em árvore que lista todos os caminhos de endereços.

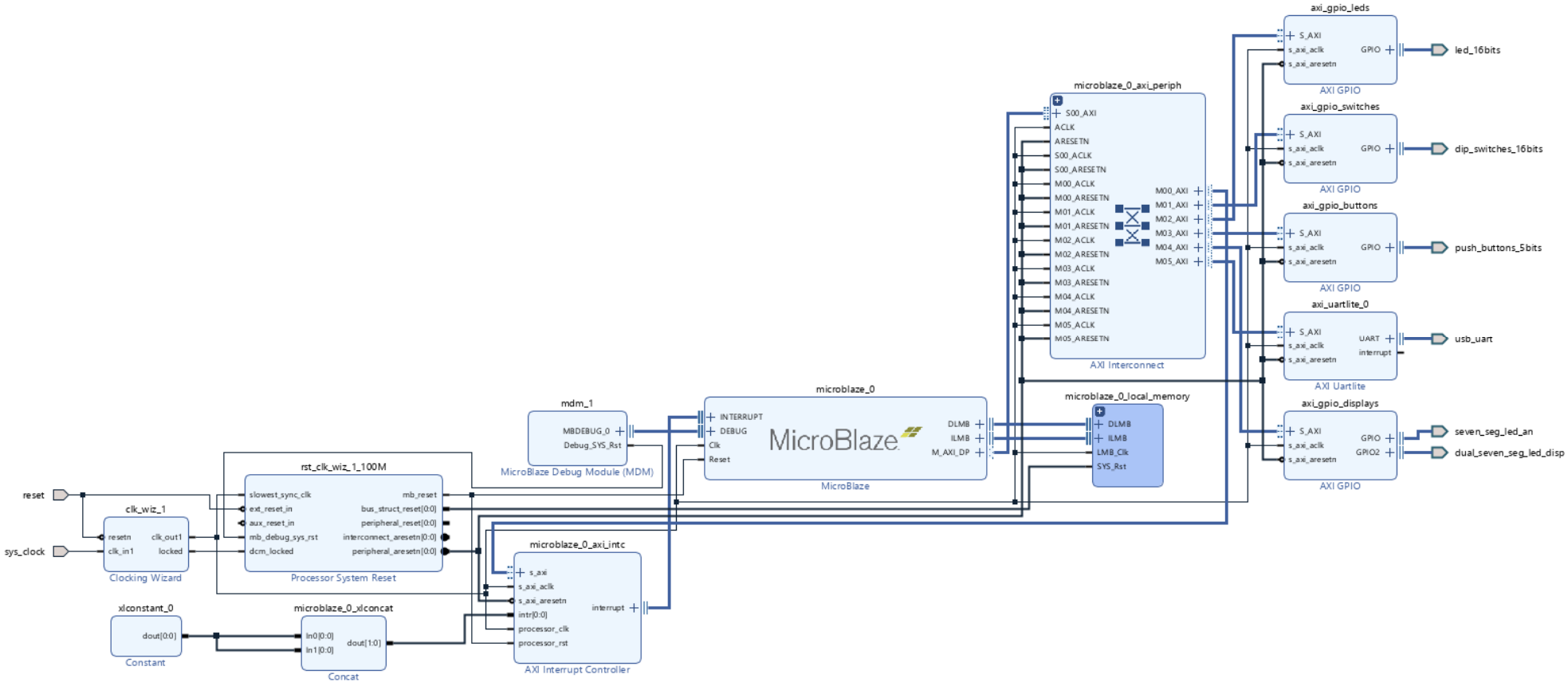
Diagram | Address Editor | Address Map

Assigned (8) Unassigned (0) Excluded (0) Hide All

Name	Interface	Slave Segment	Master Base Address	Range	Master High Address
Network 0					
/microblaze_0					
/microblaze_0/Data (32 address bits : 4G)					
/axi_gpio_buttons/S_AXI	S_AXI	Reg	0x4002_0000	64K	0x4002_FFFF
/axi_gpio_displays/S_AXI	S_AXI	Reg	0x4003_0000	64K	0x4003_FFFF
/axi_gpio_leds/S_AXI	S_AXI	Reg	0x4001_0000	64K	0x4001_FFFF
/axi_gpio_switches/S_AXI	S_AXI	Reg	0x4000_0000	64K	0x4000_FFFF
/axi_uartlite_0/S_AXI	S_AXI	Reg	0x4060_0000	64K	0x4060_FFFF
/microblaze_0_axi_intc/S_AXI	s_axi	Reg	0x4120_0000	64K	0x4120_FFFF
/microblaze_0_local_memory/dlmb_bram_if_cntlr/SLMB	SLMB	Mem	0x0000_0000	64K	0x0000_FFFF
Network 1					
/microblaze_0					
/microblaze_0/Instruction (32 address bits : 4G)					
/microblaze_0_local_memory/ilmb_bram_if_cntlr/SLMB	SLMB	Mem	0x0000_0000	64K	0x0000_FFFF

Universidade de Aveiro

Design de Bloco (BD)



Universidade de Aveiro

Implementando Hardware

Validar design

- você pode executar uma verificação abrangente do projeto.

Gerar produtos de saída (Global)

- Após a conclusão do BD e a validação do projeto, você deve gerar produtos de saída para síntese e simulação, para integrar o BD em um projeto RTL de nível superior. Os arquivos fonte e as restrições apropriadas para todo o IP são gerados e disponibilizados na janela Fontes do Vivado® Design Suite (IDE).

- A geração dos produtos de saída gera a netlist de nível superior do BD. A netlist é gerada na linguagem HDL especificada pelo Configurações → Geral → Idioma alvo do projeto.

Criar wrapper HDL

- Este comando gera um arquivo HDL de nível superior com um modelo de instanciação para o integrador IP BD.

Gerar fluxo de bits

Problemas e Resultados

Abra o design sintetizado e execute os seguintes comandos TCL:

- set_property CONFIG_VOLTAGE 3.3 [get_designs synth_1]
- set_property CFGBVS VCCO [get_designs synth_1]

Em seguida, feche o design sintetizado e salve um arquivo XDC com qualquer nome

Ferramentas de desenvolvimento

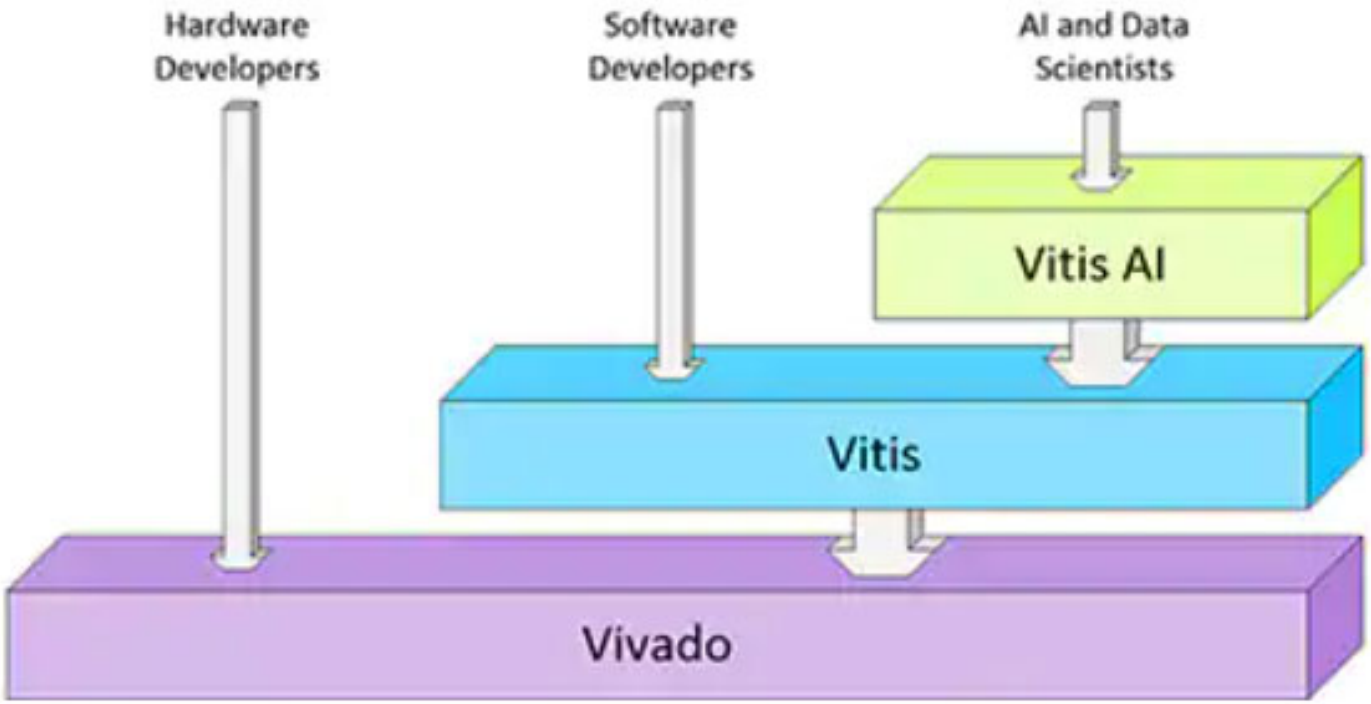


Figure 6: A high-level view of the Xilinx Vivado and Vitis design tool stack reflects how users can work with the tools at the most appropriate levels of abstraction. Hardware designers work with Vivado, software developers work with Vitis, and AI and data scientists work with Vitis AI. (Image source: Max Maxfield)

Desenvolvimento de software

Exportar hardware

Lançar Vitis

- Crie um espaço de trabalho Vitis

Criar novo projeto de aplicativo

- Crie uma nova plataforma de hardware (XSA)
- Crie um projeto “helloworld”

Dispositivo do programa

Inicie o terminal serial Vitis (ou qualquer outro aplicativo semelhante)

Construa o projeto

Execute o aplicativo de software

Universidade de Aveiro

Considerações finais

Ao final desta palestra você deverá ser capaz de:

- crie uma plataforma de hardware com MicroBlaze e GPIOs
- crie e execute um projeto Vitis simples

Pendência:

- Construa a plataforma de hardware considerada
- Teste os aplicativos fornecidos no Vitis

Universidade de Aveiro