

RISC-V_report

2021年11月30日

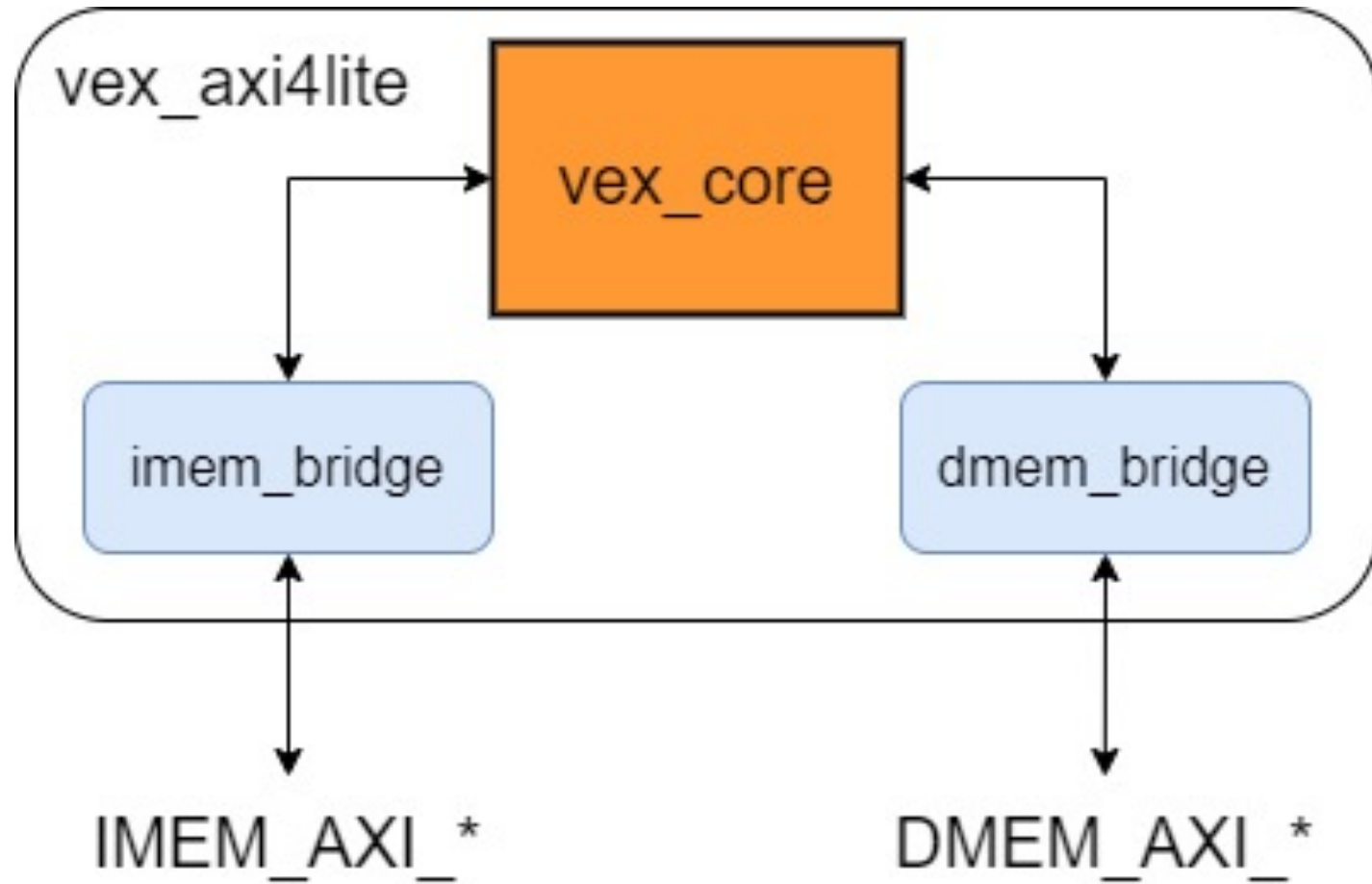
RISC-Vプロセッサ

提供されているプロセッサRTLには、次の機能を備えたFPGA対応のRISC-V ISACPU実装である、VexRiscvが含まれています。

- RV 3 2 IM命令セット
- 5段階でパイプライン化（フェッチ、デコード、実行、メモリ、ライトバック）
- FPGA用に最適化
- シングルサイクルシフター
- シングルサイクル乗数
- 動的分岐予測
- 各パイプラインステージには、バイパスまたはインターロックハザードロジックがあります

プロセッサには、命令キャッシュとデータキャッシュ、MMU、割り込み、およびデバッガがありません。

RISC-Vプロセッサのブロック図



RISC-Vプロセッサのブロック図

RISC-Vプロセッサのブロック図での
vex_axi4liteは、以下を含む最上位のモジュールです。

- VexRiscvインスタンス(vex_core)
- RISC-V命令/プログラムメモリへのAXI4Liteインターフェイスを提供する命令メモリブリッジ(imem_bridge)
- データメモリへのAXI4Liteインターフェイスを提供するデータメモリブリッジ(dmem_bridge)

このドキュメントでは、IMEMは命令メモリを指し、DMEMはデータメモリを指します。リセット後、RISC-Vプロセッサはアドレス0xA0000000からフェッチを開始します(PCをリセットします)。

RISC-Vプロセッサのインターフェース信号

トップモジュール (vex_axi4lite) インターフェース信号

信号名	入出力	説明
ACLK	input	クロック
ARESETn	input	リセット信号(0の場合はリセット)
IMEMインターフェース		
IMEM_AXI_ARVALID	output	IMEMインターフェースのAXI4Lite信号
IMEM_AXI_ARREADY	input	IMEMインターフェースのAXI4Lite信号
IMEM_AXI_ARADDR	output	命令フェッチに使用されるAXI4Liteアドレスバス(32bit)
IMEM_AXI_RVALID	input	IMEMインターフェースのAXI4Lite信号
IMEM_AXI_RREADY	output	IMEMインターフェースのAXI4Lite信号
IMEM_AXI_RDATA	input	AXI4Lite読み取りデータバス (32ビット) を介してRISC-V命令を返します。
IMEM_AXI_RRESP	input	IMEMインターフェースのAXI4Lite信号

RISC-Vプロセッサのインターフェース信号

IMEM_AXI_AWVALID	output	IMEMインターフェースのAXI4Lite信号(常に0)
IMEM_AXI_AWREADY	input	IMEMインターフェースのAXI4Lite信号(don't care)
IMEM_AXI_AWADDR	output	IMEMインターフェースのAXI4Lite信号(常に0)
IMEM_AXI_WVALID	output	IMEMインターフェースのAXI4Lite信号(常に0)
IMEM_AXI_WSTRB	output	IMEMインターフェースのAXI4Lite信号(常に0)
IMEM_AXI_WDATA	output	IMEMインターフェースのAXI4Lite信号(常に0)
IMEM_AXI_WREADY	input	IMEMインターフェースのAXI4Lite信号(don't care)
IMEM_AXI_BREADY	output	IMEMインターフェースのAXI4Lite信号(常に1)
IMEM_AXI_BVALID	input	IMEMインターフェースのAXI4Lite信号(don't care)
IMEM_AXI_BRESP	input	IMEMインターフェースのAXI4Lite信号(don't care)
DMEM interface		
DMEM_AXI_ARVALID	output	DMEMインターフェース用のAXI4Lite信号
DMEM_AXI_ARREADY	input	DMEMインターフェース用のAXI4Lite信号
DMEM_AXI_ARADDR	output	ロード命令に使用されるAXI4Liteアドレスバス(32ビット)
DMEM_AXI_RVALID	input	DMEMインターフェース用のAXI4Lite信号

RISC-Vプロセッサのインターフェース信号

DMEM_AXI_RREADY	output	DMEMインターフェース用のAXI4Lite信号
DMEM_AXI_RDATA	input	AXI4Lite読み取りデータバス（32ビット）を介してロードデータを返します
DMEM_AXI_RRESP	input	DMEMインターフェース用のAXI4Lite信号
DMEM_AXI_AWVALID	output	DMEMインターフェース用のAXI4Lite信号
DMEM_AXI_AWREADY	input	DMEMインターフェース用のAXI4Lite信号
DMEM_AXI_AWADDR	output	ストア命令に使用されるAXI4Liteアドレスバス
DMEM_AXI_WVALID	output	DMEMインターフェース用のAXI4Lite信号
DMEM_AXI_WSTRB	output	書き込みストロボのAXI4Lite信号は、ストア命令タイプによって異なります
DMEM_AXI_WDATA	output	ストアデータ用のAXI4Lite書き込みデータバス
DMEM_AXI_WREADY	input	DMEMインターフェース用のAXI4Lite信号
DMEM_AXI_BREADY	output	DMEMインターフェース用のAXI4Lite信号
DMEM_AXI_BVALID	input	DMEMインターフェース用のAXI4Lite信号
DMEM_AXI_BRESP	input	DMEMインターフェース用のAXI4Lite信号

コンパイル

- Cまたはアセンブリソースコードをコンパイルするには、RISC-V GCCコンパイラが必要です。RISC-V GCCコンパイラは、ソースからビルドすることも、ビルド済みのバイナリを利用することもできます。提供されるパッケージには、RISC-V GCCビルド済みバージョン9.2.0が含まれています。ファイルを抽出します

```
tar -xvzf riscv-gcc_9.2.0.tar.gz
```

- ファイルを抽出した後、環境パスを更新する必要があります。サンプルスクリプト（setenv.sh）は、環境を初期化するために提供されています。setenv.shを抽出されたgccディレクトリにコピーします

```
cp setenv.sh ./gcc
```

- RISC-Vコードをコンパイルする前に、setenv.shを入手して環境を初期化します。
- ソースコードをコンパイルしてhexfileを生成するために、Makefileの例が提供されています。
- ベアメタルブートコードは、start.sファイルとリンカーの例でlinker.ldとして提供されています。