

初版: 2006年7月20日
2版: 2007年3月10日
3版: 2007年5月10日
4版: 2007年5月28日
5版: 2008年8月15日

OROCHIシステム仕様書 兼 動作説明書

NAIST 中島康彦

Copyright Yasuhiko NAKASHIMA. All Rights Reserved.

注意：本仕様書の内容は、予告なしに変更されることがあります。

OROCHIシステム仕様書 兼 動作説明書
Technical Newsletter

変更履歴

	旧	新	コメント
2006/6/1 初版			
2007/3/10 2版			
2007/5/10 3版			
2007/5/28 4版	第6章 独自ピン配置	第6章 標準ピン配置	VDEC向けピン制約
2008/5/12 5版		第7章 CE77提供情報	CE77向け情報追加
2008/5/20 5版	IT3H OT3LLR2 B3NNLLR2 動作率20%	IT3HX OT3LLR2X B3NNLLR2X 動作率5%	IOセル変更 動作率変更
2008/7/4 5版	pin-#2: NC01 CK:15000ps	pin-#2: VPD CK:15151.5ps	貫通電流防止用

目次

1	関連資料	7
1.1	関連仕様書・規格	7
1.2	参考文献	8
1.3	関連ソースプログラム	8
1.4	関連ツール	9
2	構成	11
2.1	物理インタフェース	11
2.1.1	GP600M-PCI の構成	12
2.1.2	XC2V6000 と ZBT-SSRAM のインタフェース	14
2.1.3	XC2V6000 と ドータボードの物理インタフェース	14
2.1.4	XC2V6000 と Local-BUS の物理インタフェース	17
2.1.5	Local-BUS による ZBT-SSRAM の直接制御	18
2.2	論理インタフェース	18
2.2.1	外部キャッシュインタフェース	19
2.2.2	PCI ホストインタフェース	19
3	記憶	25
3.1	内蔵キャッシュ	26
3.2	外部キャッシュ	26
3.3	主記憶	27
3.4	システムコール時の動作	27
4	制御	29
4.1	コンソール入出力インタフェース	30
4.2	プロセッサ制御インタフェース	30
4.3	ハードウェアモニタ制御インタフェース	33
4.4	システムコールインタフェース	35
5	実装すべき命令および機能	43
5.1	ARM 命令	43
5.2	ARM コアに必要な機能	43
5.3	内部命令	43
5.4	VLIW 命令	43
5.5	VLIW コアに必要な機能	46
5.6	主記憶アドレス空間	47
6	専用 LSI モデルにおけるピン配置	49
6.1	HQFP240	49
6.2	QFP208	51

7 HQFP240 に関する提供情報	53
7.1 ブロック図, フロアプラン, クロック系統図	53
7.2 消費電力	54
7.3 AC 特性	57
7.4 端子情報	57
 索引	 59

目 次

2.1	GP600M-PCIの概観（下はドータボード搭載時）	11
2.2	GP600M-PCIのブロック図	12
2.3	XC2V6000専用JTAGインタフェースTP位置（回路図および半田面シルク）	13
2.4	GP600M-PCIのPCIインタフェース	14
2.5	ZBT-SSRAMのタイミングチャート	15
2.6	SSRAMインタフェースのタイミングチャート(WRITE)	16
2.7	SSRAMインタフェースのタイミングチャート(WRITEバーストモード)	17
2.8	SSRAMインタフェースのタイミングチャート(READ)	17
2.9	SSRAMインタフェースのタイミングチャート(READバーストモード)	18
2.10	プロセッサ制御インタフェースのタイミングチャート(WRITE/READ)	19
2.11	PCIブリッジローカル側インタフェースのタイミングチャート(PCIからローカルへ)	20
2.12	PCIホストによるSSRAM直接制御(WRITE)	21
2.13	PCIホストによるSSRAM直接制御(WRITEバーストモード)	21
2.14	PCIホストによるSSRAM直接制御(READ)	22
2.15	PCIホストによるSSRAM直接制御(READバーストモード)	22
2.16	各ブロック間の論理インタフェース	23
3.1	SSRAM空間の用途	25
4.1	制御レジスタ	29
5.1	実装すべきARM命令(1/2)	44
5.2	実装すべきARM命令(2/2)	45
5.3	実装すべき内部命令	46
5.4	実装すべきVLIW命令	47
6.1	HQFP240のピン配置（周辺の記号はXILINXの対応ピン）	50
6.2	QFP208のピン配置（周辺の記号はXILINXの対応ピン）	51
7.1	専用LSIのブロック図兼フロアプラン	53

表 目 次

2.1	ZBT-SSRAM 側インタフェース信号の概要	15
2.2	SSRAM インタフェース信号 (110 本) の概要	16
2.3	プロセッサ制御インタフェース信号 (40 本) の概要	18
2.4	PCI ブリッジローカル側インタフェース信号の概要	19
2.5	Local-BUS 上の論理インタフェース	24
7.1	クロック情報	53
7.2	モジュール消費電力	54
7.3	メモリ構成表	55
7.4	RAM 消費電力	56
7.5	AC 特性	57
7.6	端子情報	58

Chapter 1

関連資料

本書は、OROCHI システムの仕様および動作に関する説明書である。関連仕様書、関連規格、および、関連ソースプログラムは以下の通り。

1.1 関連仕様書・規格

- PCI XILINX コンフィグレーション仕様書 第 1.0 版, 三精システム株式会社 (2004/9/2)
… proj-gp600m-oroichi1/doc/GP600M-CAM-CONFIG.doc
- GP600M コアジェネレータ解説書第 1.0 版, 三精システム株式会社 (2005/3/31)
… proj-gp600m-oroichi1/doc/GP600M-CAM-COREGEN.pdf
- GP600M-CAM コンフィグレーション CPLD 設計仕様書, 三精システム株式会社 (2004/9/2)
… proj-gp600m-oroichi1/doc/GP600M-CAM-CPLD-SPEC-1.0.doc
- GP600M-PCI 合成配置配線手順書第 1.1 版, 三精システム株式会社 (2005/4/22)
… proj-gp600m-oroichi1/doc/GP600M-CAM-MAKE-1.1.doc
- GP600M-STEP2 FPGA ソース解説書第 1.0 版, 三精システム株式会社 (2005/3/10)
… proj-gp600m-oroichi1/doc/GP600M-CAM-OVERVIEW.doc
- GP600M-STEP2 タイミングチャート第 1 版, 三精システム株式会社 (2004/6/28)
… proj-gp600m-oroichi1/doc/GP600M-CAM-TIMING.pdf
- QuickPCILocal ブリッジ IP コア仕様書 第 1 版, 三精システム株式会社 (2003/5/20)
… proj-gp600m-oroichi1/doc/GP600M-PCI-QuickPCI-SPEC.doc
- GP600M-PCI ボード仕様書 第 1.1 版, 三精システム株式会社 (2003/11/10)
… proj-gp600m-oroichi1/doc/GP600M-PCI-SPEC-20031110.pdf
- 36Mb Pipelined and Flow Through Synchronous NBT SRAMs Rev: 1.03b, GSI Technology (2005/4)
… proj-gp600m-oroichi1/doc/GS8320Z36GT.pdf
- QL5064 User's Manual Revision B, Quick Logic (2000/8)
… proj-gp600m-oroichi1/doc/QL5064_UG_B.pdf
- 0.25 μ mCMOS ASIC CE77 シリーズデザインマニュアル FATI 198-03, 富士通 (2003)
… proj-gp600m-oroichi1/doc/fujitsu/ce77dm.pdf
- CSIM \Leftrightarrow GP600M-STEP2 インタフェース仕様書第 3 版, JST 中島康彦 (2004/8/31)
… proj-gp600m-step2/doc/GP600M-CAM-SPEC/gp600m-step2.pdf
- FR550 Series Instruction Set Manual Ver.1.1, 富士通株式会社 (2002/2)
… proj-arm/doc/fr550/FRV550-1.pdf
- FR550 シリーズ MB93551 LSI 仕様書 第 1.1a 版, 富士通株式会社 (2002/3)
… proj-arm/doc/fr550/FRV550-2.pdf

- ARM Architecture Reference Manual ARM DDI 0100E, ARM Limited (2000)
… proj-arm/doc/arm/DDI0100E_arm_arm.pdf

1.2 参考文献

- 3.3V ZERO DELAY CLOCK BUFFER IDT2309, IDT (2002/11)
… proj-gp600m-oroichi1/doc/xilinx/IDT2309.pdf
- Block Memory Generator v1.1 (2006/1/18)
… proj-gp600m-oroichi1/doc/xilinx/blk_mem_gen_ds512.pdf
- Content-Addressable Memory v5.1 (2004/11/11)
… proj-gp600m-oroichi1/doc/xilinx/cam.pdf
- Digital Clock Manager (DCM) Module (2004/8/13)
… proj-gp600m-oroichi1/doc/xilinx/dcm_module.pdf
- Dual-Port Block Memory Core v6.3 (2005/8/31)
… proj-gp600m-oroichi1/doc/xilinx/dp_block_mem.pdf
- FIFO Generator v2.3 (2006/1/11)
… proj-gp600m-oroichi1/doc/xilinx/fifo_generator_ds317.pdf
- LogiCORE FIFO Generator v2.3 User Guide (2006/1/11)
… proj-gp600m-oroichi1/doc/xilinx/fifo_generator_ug175.pdf
- Single-Port Block Memory Core v6.2 (2005/4/28)
… proj-gp600m-oroichi1/doc/xilinx/sp_block_mem.pdf

1.3 関連ソースプログラム

- ARM コンパイラ
… proj-arm/src/gcc-4.1.1.tar.gz
- ARM アセンブラ/リンカ他
… proj-arm/src/binutils-2.17.tar.gz
- uClinux ファイルシステム用 ELF バイナリ展開ツール
… proj-arm/src/elf2flt-20050308.tar.gz
- uClinux ファイルシステム (ROMFS) 作成ツール
… proj-arm/src/genromfs-0.5.1.tar.gz
- ARMulator (uClinux カーネル用デバッガ)
… proj-arm/src/gdb-5.0.tar.gz
- OROCHI 制御プログラム兼命令分解型 RTL スーパスカラシミュレータ
… proj-arm/src/isim/*
- GP600M-OROCHI FreeBSD 用ドライバ
… proj-arm/src/isim/Driver.c
- GP600M-OROCHI FreeBSD 用ドライバ組み込み手順書
… proj-arm/src/isim/Driver.hlp
- GP600M-OROCHI FreeBSD 用ユーザ関数およびヘッダ
… proj-arm/src/isim/gp600m.c
- GP600M-OROCHI Verilog 雛型
… proj-gp600m-oroichi1/fpga/step0006/*

- GP600M-OROCHI ROMFS 読み込み関数
… proj-arm/src/isim/romfs.c
- uClinux
… proj-arm/uClinux-dist.orig.tar.gz

1.4 関連ツール

- ARM ツールチェーン（主要ツールのみ列挙）
… proj-arm/bin/*
 - arm-elf-cpp: ARM C プリプロセッサ
 - arm-elf-gcc: ARM C コンパイラ
 - arm-elf-as: ARM クロスアセンブラ
 - arm-elf-ld: ARM クロスリンカ
 - arm-elf-objdump: ARM 逆アセンブラ
 - arm-elf-elf2flt: ARM ELF バイナリ展開ツール
 - genromfs: uClinux ファイルシステム (ROMFS) 生成ツール
 - arm-elf-gdb: ARMulator (uClinux カーネル用デバッガ)
- OROCHI 制御プログラム兼命令分解型 RTL スーパスカラシミュレータ (isim) 用ライブラリ
… proj-arm/lib/isim-lib/_start.o,_sub.o
- OROCHI 制御プログラム兼命令分解型 RTL スーパスカラシミュレータ (isim)
… proj-arm/src/isim/isim
- GP600M-OROCHI コンフィグレーションツール
… proj-arm/src/isim/Config
- GP600M-OROCHI 各 MCS
… proj-arm/src/isim/step*.mcs
- uClinux カーネル
… proj-arm/uClinux-dist/images/linux
- uClinux ファイルシステム
… proj-arm/uClinux-dist/images/romfs

Chapter 2

構成

OROCHI システムは、PCI ホスト、XC2V6000 搭載 PCI ボード (GP600M-PCI) , および、プロセッサ機能からなり、プロセッサ機能の実装形態に依存して、以下の3つのモデルがある。

ARM/FPGA モデル

ARM アーキテクチャを FPGA 上に実装する形態。PCI ホストおよび GP600M-PCI のみから構成される。

OROCHI/FPGA モデル

ARM+VLIW 混在実行アーキテクチャを FPGA 上に実装する形態。PCI ホストおよび GP600M-PCI のみから構成される。

OROCHI/専用 LSI モデル

ARM+VLIW 混在実行アーキテクチャを専用 LSI パッケージに実装する形態。PCI ホスト、GP600M-PCI, 専用 LSI ソケット付きドータボード、専用 LSI から構成される。

2.1 物理インターフェース

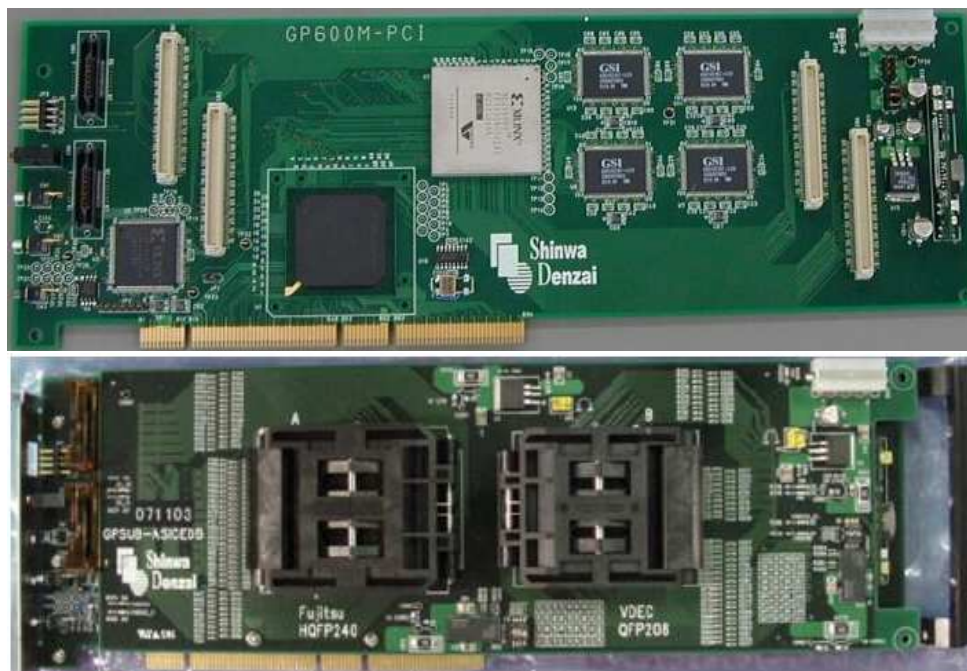


図 2.1: GP600M-PCI の概観 (下はドータボード搭載時)

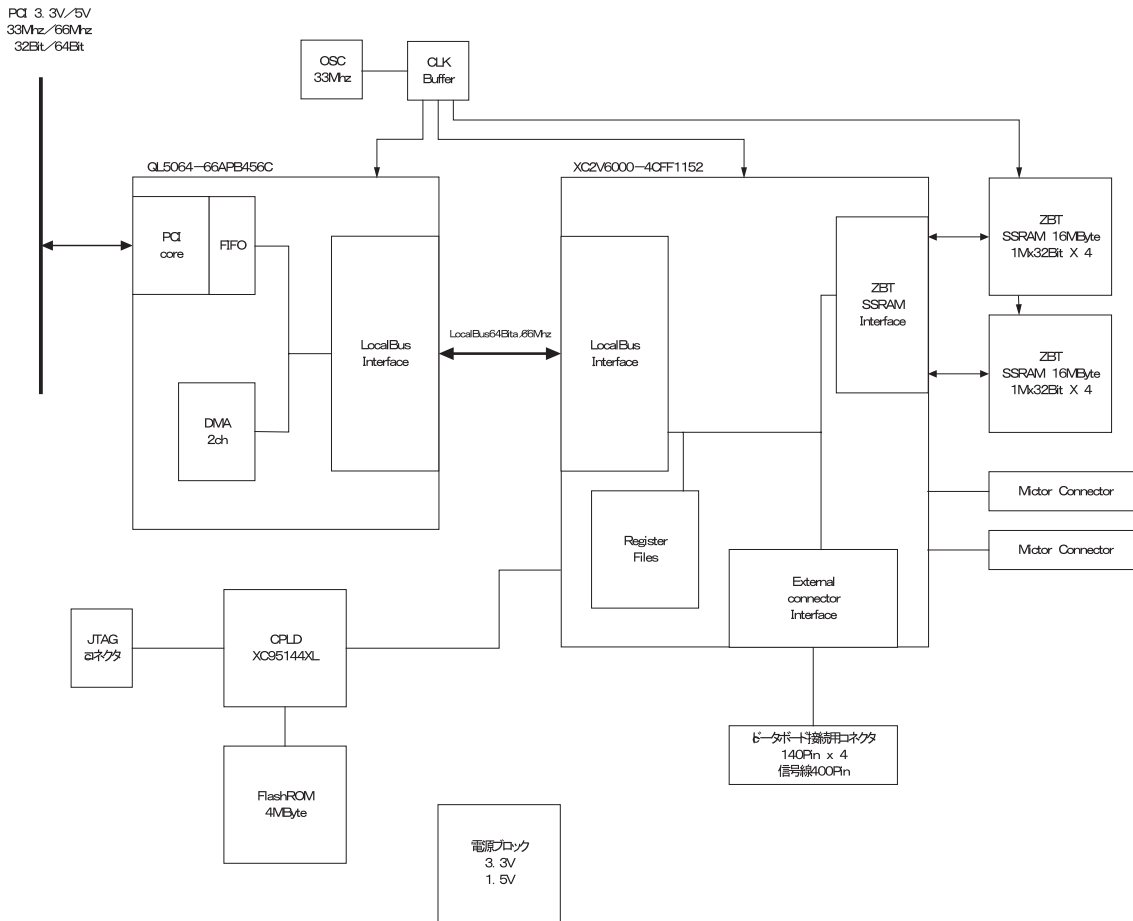


図 2.2: GP600M-PCI のブロック図

2.1.1 GP600M-PCI の構成

全体構成の理解を助けるために、図 2.1 および図 2.2 に示す GP600M-PCI の概略を説明する。なお IDT2309 から各ユニットへは 33MHz のクロックが供給されており、XC2V6000 内部の DCM モジュールにより 66MHz 等の通倍クロックを生成して使用することができる。

XC2V6000

ARM/FPGA モデルおよび OROCHI/FPGA モデルではプロセッサ機能全体が搭載される。OROCHI/専用 LSI モデルではドータボード上のプロセッサ機能に対する SSRAM インタフェースおよび PCI ホストインタフェースが搭載される。ChipScope と XC2V6000 を JTAG 接続するには、図 2.3 の JTAG 用 XTP2-5, GND および 3.3V の合計 6 本を引き出し、JTAG ケーブルに接続する。ただし、本機能の積極的な利用は推奨しない。まずは ModelSim 等による事前のシミュレーションにより十分な動作検証を行うこと。なお、以下のピンはロジックアナライザ接続用の MICTOR コネクタ (CN5, CN6)、または、デジタルオシロ接続用のタップ (ボード裏面 TP) に接続されている。

- AM2 (CN5.CLK0), AM7 (CN5.A3-7), AP6 (CN5.A3-6), AM6 (CN5.A3-5), AP5 (CN5.A3-4), AN5 (CN5.A3-3), AP4 (CN5.A3-2), AN4 (CN5.A3-1), AN3 (CN5.A3-0), AL9 (CN5.A2-7), AJ9 (CN5.A2-6), AH9 (CN5.A2-5), AN8 (CN5.A2-4), AM8 (CN5.A2-3), AL8 (CN5.A2-2), AP7 (CN5.A2-1), AN7 (CN5.A2-0)
- AM9 (CN5.CLK1), AJ11 (CN5.A1-7), AG11 (CN5.A1-6), AN10 (CN5.A1-5), AL10 (CN5.A1-4), AK10 (CN5.A1-3), AJ10 (CN5.A1-2), AP9 (CN5.A1-1), AN9 (CN5.A1-0), AG12 (CN5.A0-7), AF12 (CN5.A0-6), AE12 (CN5.A0-5), AP11 (CN5.A0-4), AN11 (CN5.A0-3), AM11 (CN5.A0-2), AL11 (CN5.A0-1), AK11 (CN5.A0-0)

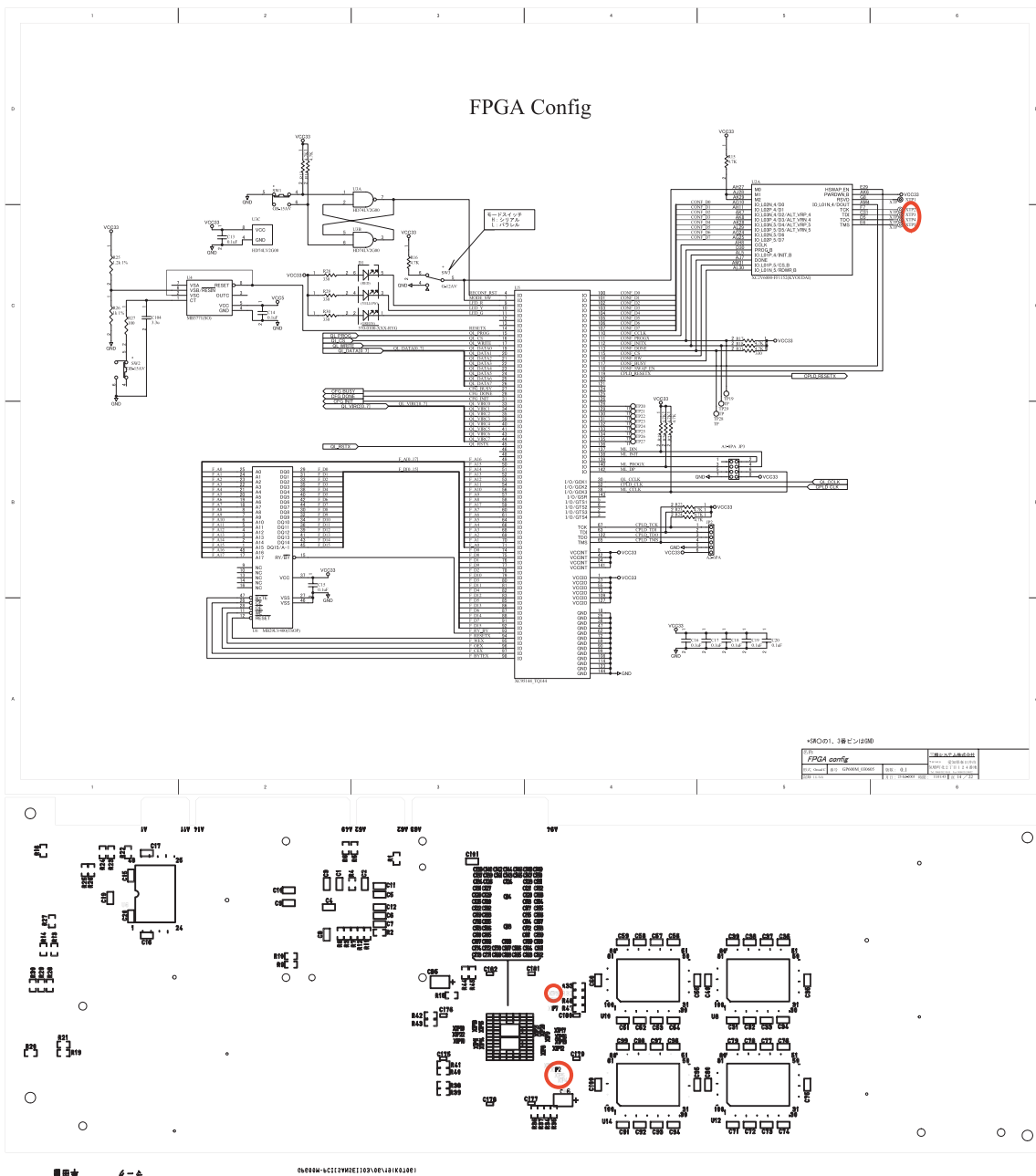


図 2.3: XC2V6000 専用 JTAG インタフェース TP 位置 (回路図および半田面シルク)

- AH12 (CN6.CLK0), AH13 (CN6.A3-7), AG13 (CN6.A3-6), AF13 (CN6.A3-5), AE12 (CN6.A3-4), AP12 (CN6.A3-3), AN12 (CN6.A3-2), AM12 (CN6.A3-1), AL12 (CN6.A3-0), AG14 (CN6.A2-7), AF14 (CN6.A2-6), AE14 (CN6.A2-5), AP13 (CN6.A2-4), AM13 (CN6.A2-3), AL13 (CN6.A2-2), AK13 (CN6.A2-1), AJ13 (CN6.A2-0)
- AK14 (CN6.CLK1), AH15 (CN6.A1-7), AG15 (CN6.A1-6), AF15 (CN6.A1-5), AE15 (CN6.A1-4), AP14 (CN6.A1-3), AN14 (CN6.A1-2), AM14 (CN6.A1-1), AL14 (CN6.A1-0), AJ16 (CN6.A0-7), AH16 (CN6.A0-6), AG16 (CN6.A0-5), AE16 (CN6.A0-4), AD16 (CN6.A0-3), AP15 (CN6.A0-2), AM15 (CN6.A0-1), AL15 (CN6.A0-0)
- A31 (TP11), B31 (TP12), B32 (TP13), C33 (TP14), C2 (TP15), B3 (TP16), B4 (TP17), A4 (TP18)

CPLD

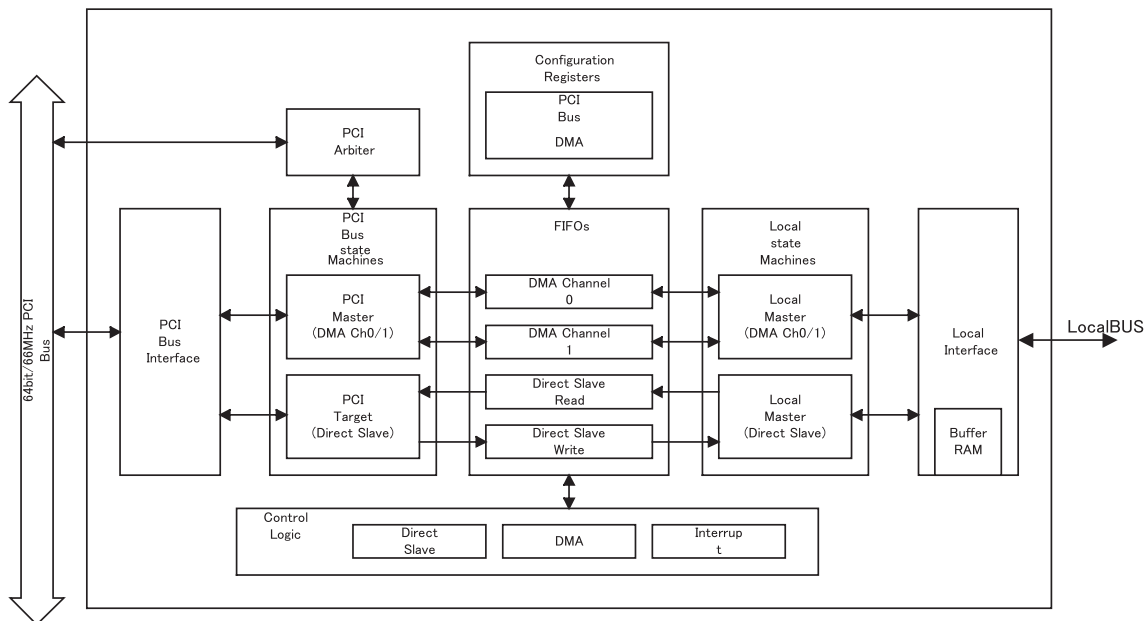


図 2.4: GP600M-PCI の PCI インタフェース

XC2V6000 コンフィグレーション制御用の PLD. コンフィグレーション用コネクタ (JP3) またはフラッシュROM からのコンフィグレーション機能を提供する. ただし通常は PCI ホストからのコンフィグレーションを行うため, JP3 は使用しない. JTAG コネクタ (JP2) は PLD 専用につきユーザは使用不可.

ZBT-SSRAM

GSI 社の GS8320Z36. 36bit 幅× 4M ワードを 2 組搭載. 並置し, 64bit 幅× 4M ワードの外部キャッシュメモリ機構として使用する. プロセッサ機能と外部キャッシュメモリのインタフェースのための制御情報も格納する.

PCIブリッジ機能 (QL5064)

PCI64Bit/66Mhz インタフェースのためのブリッジ機能. PCIバスマスター機能を有し, 独立する 2ch の DMA を搭載. 図 2.4 に詳細を示す.

Local-BUS

PCIブリッジと XC2V6000 を接続する汎用バス. 詳細は QuickPCILocal ブリッジ IP コア仕様書 第 1 版を参照.

ドータボード接続コネクタ

GP600M-PCI には, ドータボード用拡張 I/O ピンが装備されている. 計 4 個の高密度コネクタによりドータボードと接続可能. 各 140 ピン中 I/O は 100 ピン (I/O 合計 400 ピン). I/O ピン番号は共通で, 11-20, 23-32, 35-44, 47-56, 59-68, 71-80, 83-92, 95-104, 107-116, 119-128.

2.1.2 XC2V6000 と ZBT-SSRAM のインタフェース

GP600M-PCI は 36bit × 1M ワードの ZBT-SSRAM を計 8 個搭載しており, 36bit × 4M ワード × 2 組の外部メモリとして使用できる. 表 2.1 にインタフェース信号の概要, 図 2.5 にタイミングチャートを示す.

2.1.3 XC2V6000 とドータボードの物理インタフェース

XC2V6000 とドータボードの物理インタフェースは, OROCHI/専用 LSI モデルでは, ドータボード上のプロセッサに対してメモリ機能を提供する SSRAM インタフェース (110 本), および, プロセッサ制御インタフェース (40 本) を収容する. 表 2.2 に SSRAM インタフェース信号の概要, 図 2.6 から図 2.9 に

表 2.2: SSRAM インタフェース信号 (110 本) の概要

信号名	説明
M_XREQ(入力)	SSRAM 参照要求信号, Active-LOW
M_XGNT(出力)	要求受付完了信号, Active-LOW
M_WEX(入力)	書き込みイネーブル, Active-LOW
M_BSTMX(入力)	LOW はバーストモード
M_WRDY(出力)	書込完了信号, Active-HIGH. 本信号の ON をもってプロセッサは参照要求を消去できる
M_RRDY(出力)	読出データ valid 信号, Active-HIGH. 本信号の ON をもってプロセッサは参照要求を消去できる
M_A(入力)	アドレス, A31-3
M_BEX(入力)	バイト毎書き込みフラグ, Active-LOW, B7-0
M_D(入出力)	データ, D63-0
M_BSYX(出力)	SSRAM 動作中を示すビジー信号, Active-LOW
M_STAT(出力)	状態表示信号, STAT1-0 (0:empty 2:OP-ok 3:IF-ok)

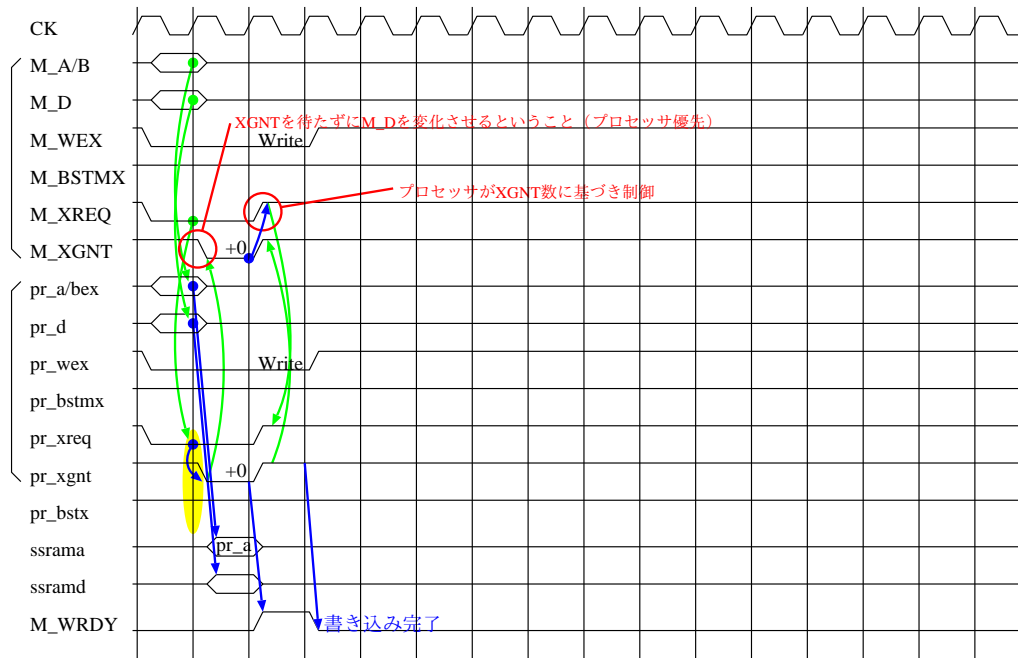


図 2.6: SSRAM インタフェースのタイミングチャート (WRITE)

トである。

表 2.3 にプロセッサ制御インタフェース信号の概要, 図 2.10 にタイミングチャートを示す. XINT は, PCI ホストが後述する BAR2 の先頭ワードへの書き込みを完了し, プロセッサに対して動作を指示する際に ON になる. プロセッサが XINT を検出し XACK を ON にした時点で, XINT は OFF にならなければならない. XREQ はプロセッサが BAR2 に対して書き込みまたは読み出しを行う際に ON にする信号線である.

専用 LSI に対し, JTAG によるバウンダリスキャン機能をどうするかについては, ハードウェアインタフェースや制御ソフトウェアも含めて全て未定

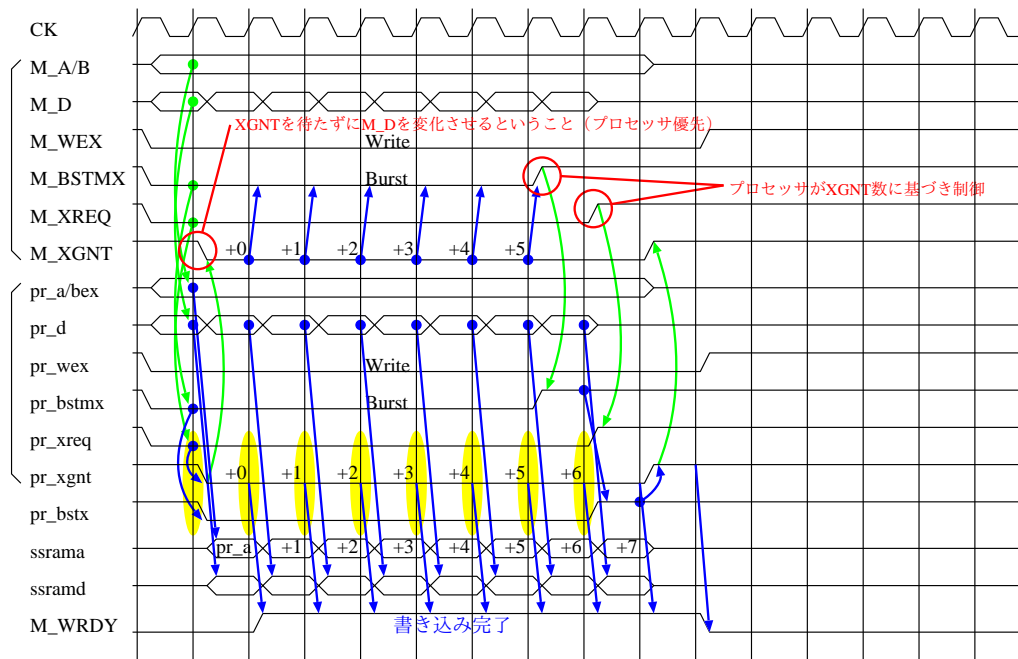


図 2.7: SSRAM インタフェースのタイミングチャート (WRITE バーストモード)

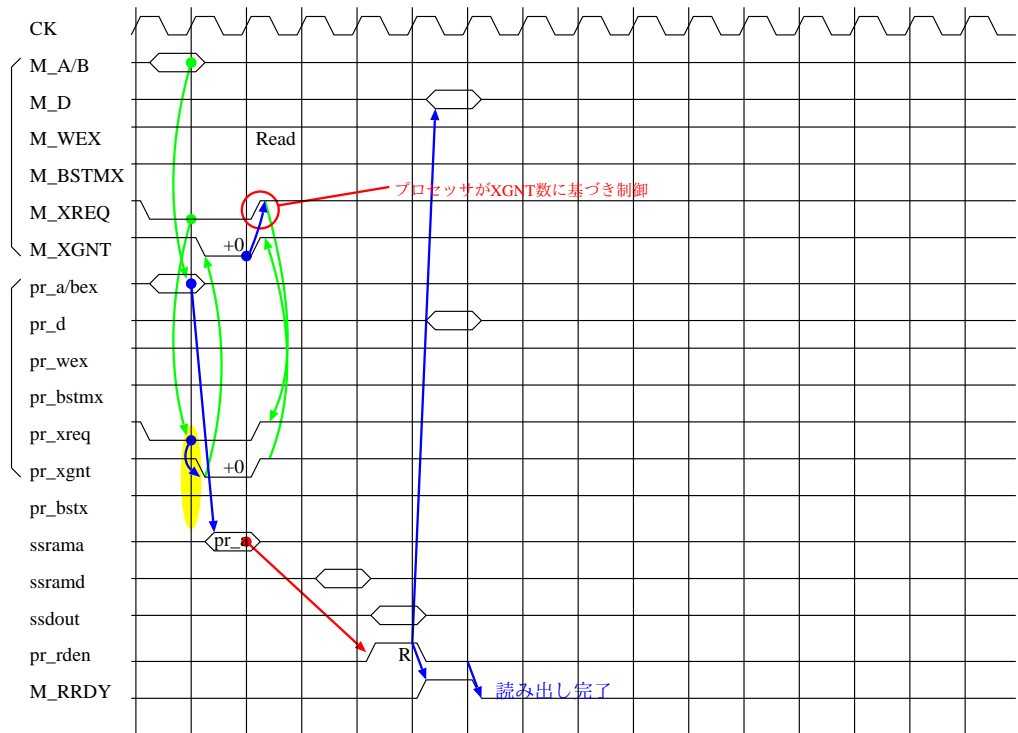


図 2.8: SSRAM インタフェースのタイミングチャート (READ)

2.1.4 XC2V6000 と Local-BUS の物理インタフェース

GP600M-PCI と PCI ホストは、前述の PCI ブリッジにより接続される。表 2.4 にインタフェース信号の概要、図 2.11 にタイミングチャートを示す。

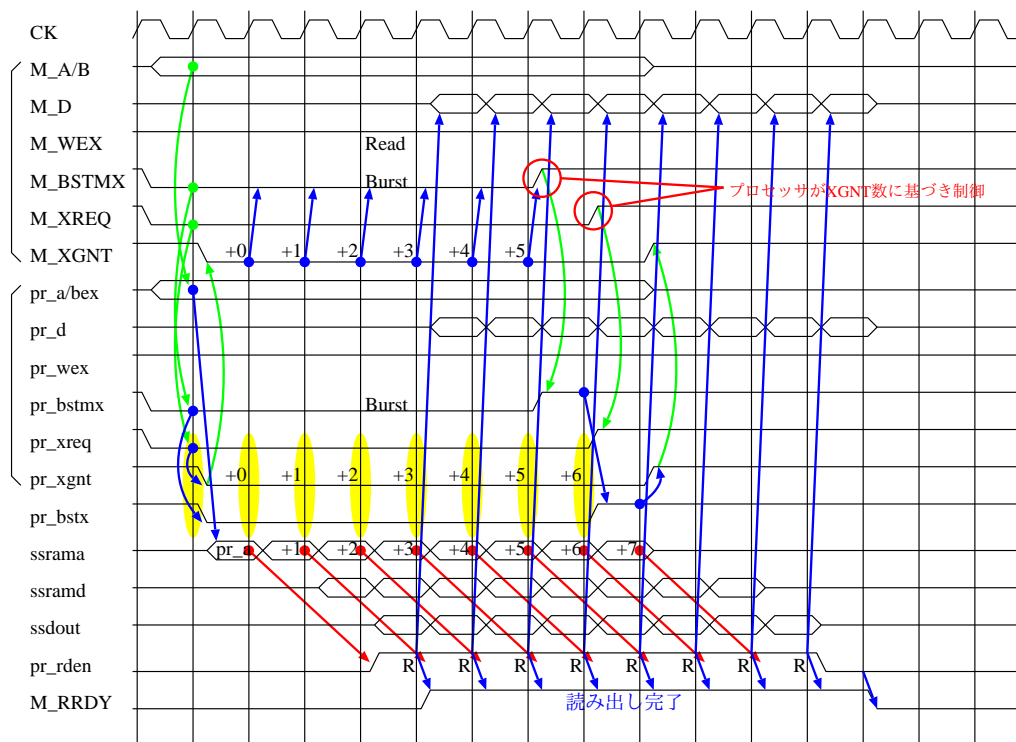


図 2.9: SSRAM インタフェースのタイミングチャート (READ バーストモード)

表 2.3: プロセッサ制御インタフェース信号 (40 本) の概要

信号名	説明
C_XINT(出力)	PCI ホストからのリクエスト信号, Active-LOW
C_XACK(入力)	プロセッサからの INT 受付け信号, Active-LOW
C_XREQ(入力)	プロセッサからのレジスタ参照信号, Active-LOW
C_A(入力)	アドレス, A3-0
C_WEX(入力)	書き込みイネーブル, Active-LOW
C_D(入出力)	データ, D31-0

2.1.5 Local-BUS による ZBT-SSRAM の直接制御

前述の ZBT-SSRAM および Local-BUS インタフェースの組み合わせにより, Local-BUS から SSRAM を直接制御できる. すなわち, PCI ホストの主記憶空間に ZBT-SSRAM を対応付け, PCI ホストが直接 ZBT-SSRAM の内容を参照/更新することができる. 図 2.12 から図 2.15 に, Loca-BUS による SSRAM 直接制御を示す.

2.2 論理インタフェース

図 2.16 にプロセッサと外部との論理的な接続を示す. このうち, GP600M-PCI と PCI ホストの論理インタフェースは, PCI-64bit/66MHz バス上のレジスタ空間に写像された GP600M-PCI 上のレジスタ群 BAR0-3 である (表 2.5). BAR0-2 を `ioctl()` 経由, BAR3 を `mmap()` によりユーザプログラム内配列に写像して使用するか, または, BAR3 の後半アドレスに BAR2 の機能を割り当て, 全てを `mmap()` により写像して使用する. `ioctl()` のオーバーヘッドを削減する場合は後者が適切である. プロセッサは, 本レジスタ空間に対応付けられた次の 2 種類の外部装置に接続される.

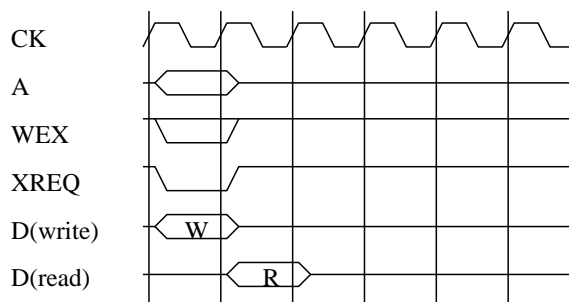


図 2.10: プロセッサ制御インタフェースのタイミングチャート (WRITE/READ)

表 2.4: PCI ブリッジローカル側インタフェース信号の概要

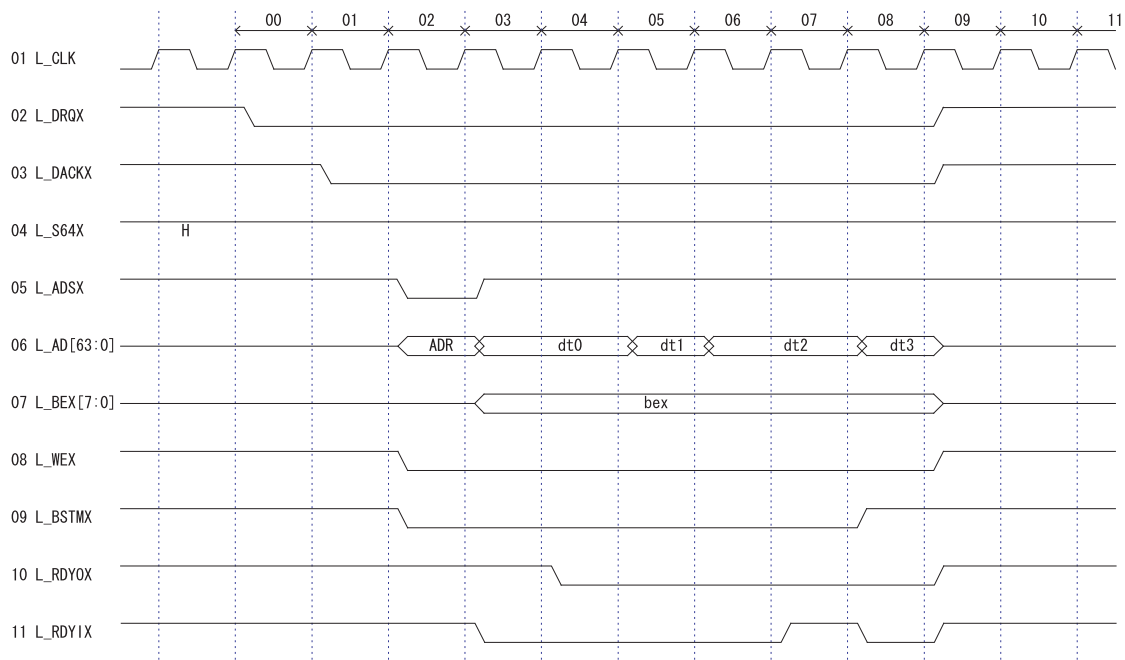
信号名	説明
L_RSTX(出力)	リセット信号
L_DRQX(出力)	ブリッジからの転送要求信号
L_DACKX(入力)	転送要求に対する応答 (LOW の場合ローカル側がスレーブ, HIGH の場合ローカル側がマスタ)
L_ADSX(入出力)	転送開始 (バスマスタが出力)
L_WEX(入出力)	LOW は WRITE, HIGH は READ に対応
L_BSTMX(入出力)	LOW はバーストモードに対応
L_RDYOX(出力)	ブリッジ側転送準備 OK
L_RDYIX(入力)	ローカル側転送準備 OK
L_AD(入出力)	ブリッジ側マスタの時はアドレス+データ, ローカル側マスタの時はアドレス+バイト数+データ
L_BEX(入出力)	バイトイネーブル
L_INTX(入力)	割り込み信号
L_S64X(入力)	PCI 制御レジスタ選択信号
L_DBSYX(出力)	PCI 書き込み状態の表示 (ただし, 特に使用しなくてよい)

2.2.1 外部キャッシュインタフェース

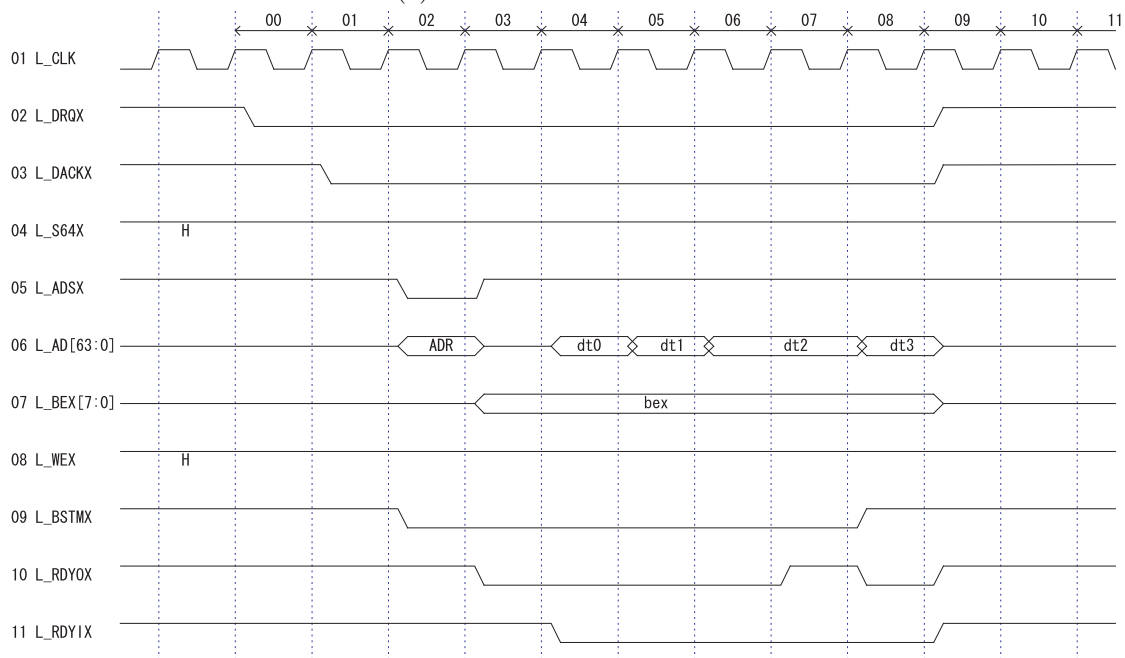
プロセッサ機能に対して, 外部キャッシュインタフェースを提供する. 詳細は第 3 章において説明する.

2.2.2 PCI ホストインタフェース

プロセッサ機能に対して, PCI ホストからのプロセッサ制御機能, システムコール (SWI 命令) の PCI ホスト実行機能, コンソール入出力機能, スキャン機能を提供する. 詳細は第 4 章において説明する.



(1) PCI ⇒ローカルの書き込み



(2) PCI ⇒ローカルの読み出し

図 2.11: PCIブリッジローカル側インタフェースのタイミングチャート (PCIからローカルへ)

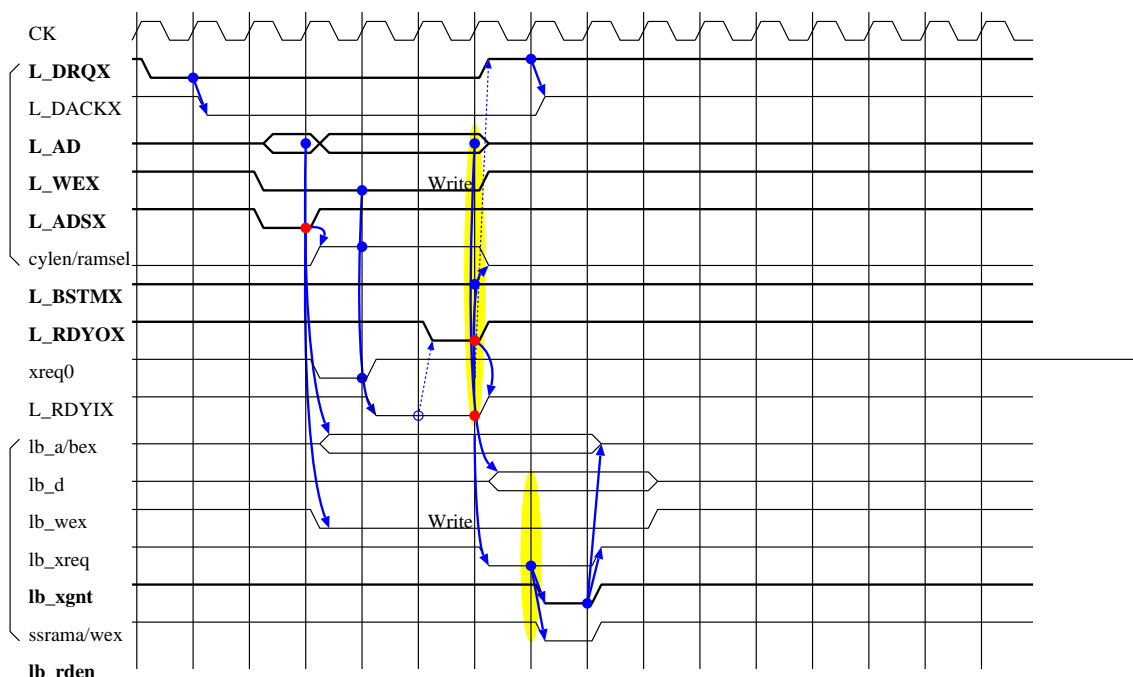


図 2.12: PCI ホストによる SSRAM 直接制御 (WRITE)

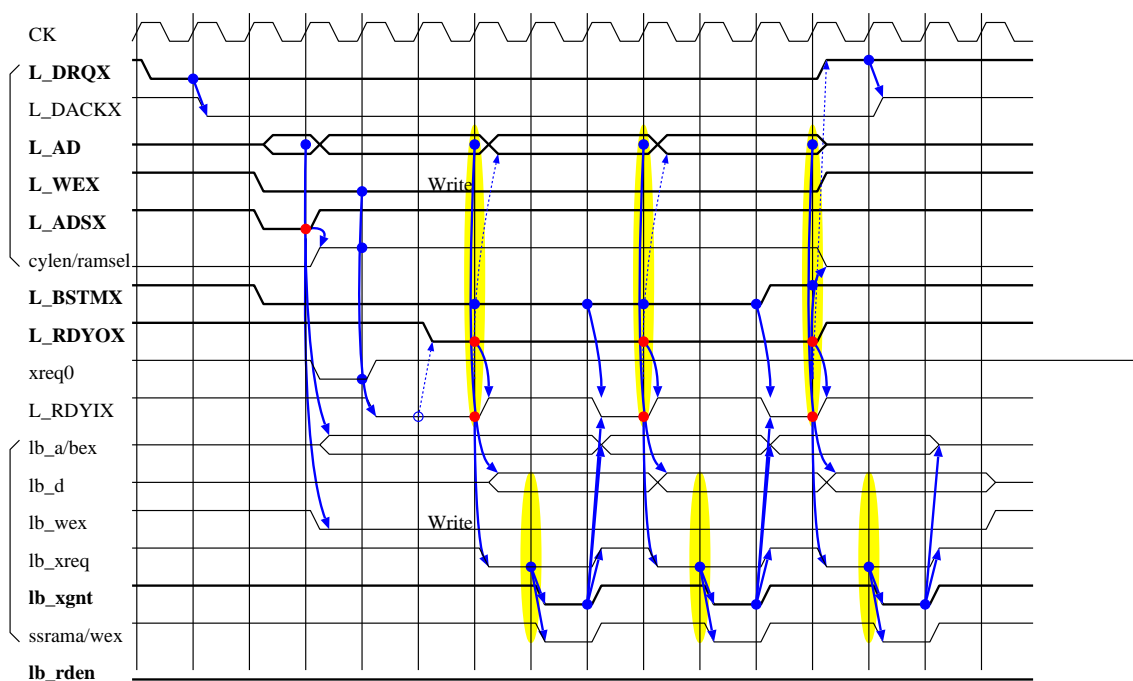
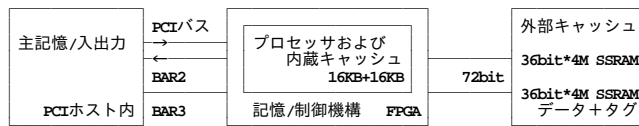
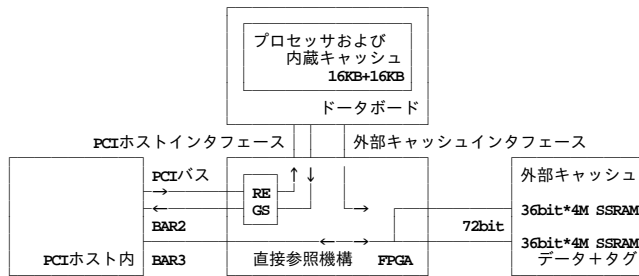


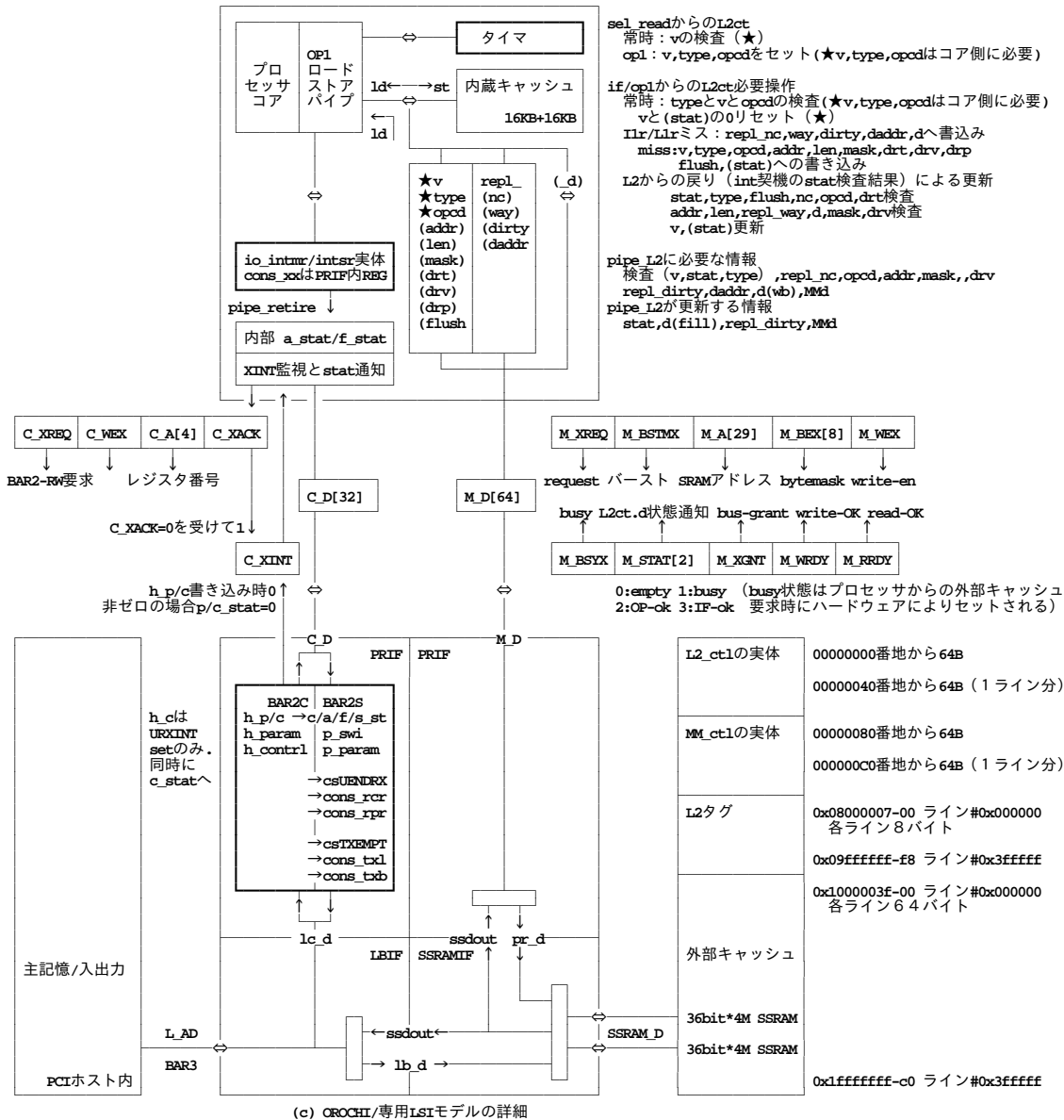
図 2.13: PCI ホストによる SSRAM 直接制御 (WRITE バーストモード)



(a) ARM/FPGA, OROCHI/FPGAモデル



(b) OROCHI/専用LSIモデル



(c) OROCHI/専用LSIモデルの詳細

図 2.16: 各ブロック間の論理インタフェース

表 2.5: Local-BUS 上の論理インタフェース

atop(vtophys((vm_offset_t)rman_get_virtual(bus_alloc_resource(dev, SYS_RES_MEMORY, RID#))))		
インタフェース名	RID#	説明
BAR0(0x0-0xff)	PCI_CBMA0:0x10	PCI 制御用 QL5064 内レジスタ
BAR1(0x0-0xff)	PCI_CBMA1:0x14	PCI 制御用 QL5064 内レジスタ (XC2V6000 コンフィグレーション用)
BAR2(0x0-0x1ff)	PCI_CBMA2:0x18	ユーザ I/O. PCI ホストがプロセッサ機能へ動作を指示する際、および、プロセッサ機能から PCI ホストへの例外通知/サービス要求の際に使用する。
BAR3(0x0-0x3ffff)	PCI_CBMA3:0x1C	前半 32MB は SSRAM 空間、後半 32MB は未使用である。外部キャッシュとして使用する SSRAM の内容を PCI ホストから直接参照する際に使用する。

Chapter 3

記憶

BAR3内アドレス	説明	各ビットの説明		備考
0x000007 - 00	L2_ctl L2制御情報		bit 1- 0 : v bit 5- 4 : stat bit 7 : flush bit15- 8 : t bit19-16 : type bit23-20 : opcd bit31-24 : len (1/2)	processorから3書込時はstatに1を指定 processorから0書込時, M_STAT無変更時はstatに3を指定 M_STAT=0変更時はstatに0を指定 hostからstatを更新する際にvも変更されるがprocessorはvを参照しない (内部的なvを参照するのみ) processorから3書込時はM_STATをコピー hostからの場合はそのままM_STATに伝搬
0x000000f - 08	L2_ctl L2制御情報	bit63-32 : addr	bit31- 0 : mask	
0x0000017 - 10	L2_ctl L2制御情報	bit63-32 : drv	bit 1- 0 : drt bit27-16 : drp	
0x000001f - 18	L2_ctl L2制御情報	bit63-32 : repl_daddr	bit 0 : repl_nc bit 1 : repl_dirty bit31-24 : repl_way	
0x000003f - 20	未使用			
0x000007f - 40	L2_ctl 1-line分64Bバッファ	byte63-32 : dline	byte31- 0 : dline	
0x0000087 - 80	MM_ctl 主記憶制御情報		bit 1- 0 : v bit 5- 4 : stat bit15- 8 : t bit19-16 : type bit23-20 : opcd bit31-24 : len (1/16)	0:empty 1:reserve 3:inuse 0:empty 2:MMct-ok timer 8:ld,9:st
0x000008f - 88	MM_ctl 主記憶制御情報	bit63-32 : addr	bit31- 0 : mask	mask for single-word
0x0000097 - 90	MM_ctl 主記憶制御情報	bit63-32 : repl_daddr	bit 1 : repl_dirty bit31-24 : pid	L2-dirty target pid
0x00000bf - 98	未使用			
0x00000ff - c0	MM_ctl 1-line分64Bバッファ	byte63-32 : dline	byte31- 0 : dline	
0x0000100 : 0x07ffff	未使用			
0x0800003 - 00 : 0x08ffff - fc	外部キャッシュタグ #0x00000 : 外部キャッシュタグ #0x3ffff	bit 0 : valid bit 1 : dirty bit31-24 : ltag	bit 0 : valid bit 1 : dirty bit31-24 : ltag	キャッシュタグ本体 物理アドレス上位4bit
0x0a00000 : 0x0ffffff	未使用			
0x100003f - 00 : 0x1ffffff - c0	外部キャッシュライン#0x00000 : 外部キャッシュライン#0x3ffff	byte63-32 : dline	byte31- 0 : dline	キャッシュライン本体
0x2000000 : 0x20001ff	BAR2 制御レジスタ空間			

図 3.1: SSRAM 空間の用途

アドレス計算は 32 ビットでラップ・アラウンドする。2,4,8 バイトをそれぞれ半語、語、倍長語と呼ぶ。半語、語、倍長語は任意のアドレスから始まるバイト位置に置けるのではなく、決まったアドレスの場所の

みに置くことができる。すなわち、0番地、または、それぞれ2,4,8の倍数のアドレスから始まる場所にしか置けない。レジスタの各ビットとの関係はリトルエンディアンに従う。また、アドレスのタイプは実アドレスである。

キャッシュابل主記憶アドレス空間の読み出しは、プロセッサ内蔵キャッシュ、外部キャッシュ、PCIホスト主記憶装置の優先順に行われる。主記憶アドレス空間への書き込みは、内蔵キャッシュから外部キャッシュへはストアスワップ、外部キャッシュから主記憶へもストアスワップ方式に従う。各記憶階層間のインタフェース領域および外部キャッシュは、図 3.1 に示す SSRAM 空間に配置される。

3.1 内蔵キャッシュ

1次命令キャッシュおよびデータキャッシュは、各々4ウェイの物理アドレスキャッシュ、ラインサイズは64バイト、容量は各16Kバイトである。

内蔵キャッシュの読み出しに際してキャッシュミスが検出された場合、最終的に外部キャッシュから転送される有効ラインの格納先となる入れ換え対象ラインが4-wayの中からLRUアルゴリズムに基づいて決定される。入れ換え対象ラインがダーティラインである場合、入れ換え対象ラインの内容が外部キャッシュインタフェース内バッファ(L2_ctl.dline)に書き込まれ、外部キャッシュへの追い出しおよび有効ラインの読み出し要求が、追い出し先主記憶アドレス、読み出し主記憶アドレス、書き込み先ウェイ番号(L2_ctl.repl_daddr, L2_ctl.addr, L2_ctl.repl_way)とともに外部キャッシュインタフェースにキューイング(L2_ctl.v=3, stat=1, M_STAT=1, type)される。一方、入れ換え対象ラインがダーティラインでない場合、有効ラインの読み出し要求が、読み出し主記憶アドレスとともに外部キャッシュインタフェースにキューイング(L2_ctl.v=3, stat=1, M_STAT=1, type)される。外部キャッシュから到着した有効ラインは一旦外部キャッシュインタフェース内バッファ(L2_ctl.dline)に格納される。プロセッサ機能は、M_STAT=2/3による通知を受けて、内蔵キャッシュの入れ換え対象ラインに書き込む。

内蔵キャッシュへの書き込みに際して内部キャッシュミスが検出された場合も同様にキューイングされる。外部キャッシュから到着した有効ラインは一旦外部キャッシュインタフェース内バッファ(L2_ctl.dline)に格納される。プロセッサ機能は、M_STAT=2/3による通知を受けて、内蔵キャッシュの入れ換え対象ラインに、ストアデータとともに書き込む。また、入れ換え対象ラインのダーティビットをセットする。

なお、プロセッサ機能が命令フェッチやロード命令を投機実行する場合、外部キャッシュインタフェースに一旦キューイング(L2_ctl.v=3, stat=1, M_STAT=1, type)された読み出し要求が、プロセッサ機能により消去(L2_ctl.opcd=15またはL2_ctl.type=8の場合に、L2_ctl.v=0, stat=3指定によりstat=M_STAT)されることがある。この操作によっても、外部キャッシュの動作が停止することは保証されず、後述の動作によりM_STAT=2/3が報告されることがある。プロセッサ機能が、読み出し要求を消去した後に、外部キャッシュインタフェースに対して新たな要求をキューイングする際には、M_STATがbusy(=1)でないことを確認しなければならない。

3.2 外部キャッシュ

命令およびデータ共用外部キャッシュは、ダイレクトマップの物理アドレスキャッシュ、ラインサイズは64バイト、容量は16Mバイトである。

外部キャッシュインタフェースに読み出し要求がキューイング(L2_ctl.v=3, stat=1, M_STAT=1, type)されている場合、外部キャッシュが検査される。キャッシュヒットの場合、当該ラインの内容が外部キャッシュインタフェース内バッファ(L2_ctl.dline)に格納され、内蔵キャッシュに対して有効ラインの到着がM_STAT=2/3により通知される。外部キャッシュミスが検出された場合で、入れ換え対象ライン(ダイレクトマップなので唯一に決まる)がさらにダーティラインである場合、入れ換え対象ラインの内容が主記憶インタフェース内バッファ(MM_ctl.dline)に読み出され、主記憶への追い出しおよび有効ラインの読み出し要求が、追い出し先主記憶アドレスおよび読み出し主記憶アドレスとともに主記憶インタフェース(MM_ctl.repl_daddr, MM_ctl.addr)にキューイングされる。一方、入れ換え対象ラインがダーティライン

でない場合、有効ラインの読み出し要求が、読み出し主記憶アドレスとともに主記憶インタフェースにキューイングされる。

主記憶から到着した有効ラインは一旦主記憶インタフェース内バッファ (MMctl.dline) に格納され、外部キャッシュの入れ換え対象ラインに書き込まれるとともに、外部キャッシュインタフェース内バッファ (L2ctl.dline) に格納され、内蔵キャッシュに対して有効ラインの到着が M_STAT=2/3 により通知される。

外部キャッシュインタフェースに追い出し要求がキューイングされている場合、追い出し先となる外部キャッシュが検査される。外部キャッシュミスが検出された場合で、入れ換え対象ライン (ダイレクトマップなので唯一に決まる) がさらにダーティラインである場合、入れ換え対象ラインの内容が主記憶インタフェース内バッファ (MMctl.dline) に読み出され、主記憶への追い出し要求が、追い出し先主記憶アドレスとともに主記憶インタフェースにキューイングされる。一方、入れ換え対象ラインがダーティラインでない場合、または、外部キャッシュヒットの場合、外部キャッシュインタフェース内バッファ (L2ctl.dline) の内容が入れ換え対象ラインに格納され、入れ換え対象ラインのダーティビットがセットされる。

なお、前章において説明した通り、SSRAM 空間は PCI ホストから直接参照/更新することができる。外部キャッシュインタフェースを含む外部キャッシュの制御は、ソフトウェアとして PCI ホスト上に実装してもよいし、ハードウェアとして XC2V6000 上に実装してもよい。1 次キャッシュミスに要する実サイクル数を減らす必要がある場合には、後者の構成が望ましい。

3.3 主記憶

主記憶を管理する PCI ホストは、主記憶インタフェースに読み出し要求がキューイングされている場合、主記憶から該当ラインを読み出し、主記憶インタフェース内バッファ (MMctl.dline) に格納した後に外部キャッシュ制御部に対して有効ラインの到着を通知する。また、主記憶インタフェースに追い出し要求がキューイングされている場合、主記憶インタフェース内バッファ (MMctl.dline) から主記憶に書き込みを行った後に外部キャッシュ制御部に対して書き込み完了を通知する。

3.4 システムコール時の動作

プロセッサが SWI 命令 (システムコール) を検出し PCI ホストに実行させる場合、PCI ホストに通知する前にプロセッサが内蔵キャッシュの内容をフラッシュし、最新データが外部キャッシュまたは主記憶上に格納されている状態を保証しなければならない。SWI 命令による停止を検出した PCI ホストは、必要に応じて外部キャッシュのダーティラインを主記憶装置に書き出し、外部キャッシュの該当ラインを無効化した上でシステムコールを実行し、プロセッサに対して継続実行要求またはリスタート要求 (longjump 時のみ) を発行しなければならない。

Chapter 4

制御

BAR2内アドレス	説明		プロセッサR/W	PCIホストR/W
コンソール割り込み制御/プロセッサ制御レジスタ				
0x0000007 - 00	h_c	bit 3- 0	R	W
	h_p	bit15- 4	R	W
	h_s	bit31-16	R	W
0x000000f - 08	h_param	下位 4 バイト	R	W
0x0000017 - 10	h_control	下位 4 バイト	R	W
0x0000027 - 20	pe0_rstcount	bit15- 0	—	W
コンソール制御/システムコール制御レジスタ				
0x0000047 - 40	c_stat	bit 3- 0	W	R (h_c非0書込時reset)
	a_stat	bit 9- 4	W	R (h_arm非0書込時busy)
	f_stat	bit15-10	W	R (h_frv非0書込時busy)
	s_stat	bit23-16	W	R (h_s非0書込時busy)
	p_swi	bit31-24	W	R
0x000004f - 48	p_param	下位 4 バイト	W	R
0x0000057 - 50	cons_csrUSENDRX	下位 4 バイト	※—	※W
0x000005f - 58	cons_rcr	下位 4 バイト	RW	RW
0x0000067 - 60	cons_rpr	下位 4 バイト	RW	RW
0x000006f - 68	cons_csrTXEMPTY	下位 4 バイト	※RW	※RW
0x0000077 - 70	cons_txl	下位 4 バイト	RW	RW
0x000007f - 78	cons_txb	下位 4 バイト	RW	RW

※TXEMPTYを更新するには、cons_csrTXEMPTYへ書き込む。
 USENDRXを更新するには、cons_csrUSENDRXへ書き込む。
 cons_csrTXEMPTYから読み出される値はTXEMPTY情報とUSENDRX情報の論理和となる。
 cons_csrUSENDRXから読み出される値は不定である。

図 4.1: 制御レジスタ

本章では、PCI ホストインタフェースについて説明する。PCI ホストインタフェースは、コンソール入出力インタフェース、プロセッサ制御インタフェース、ハードウェアモニタ制御インタフェース、システムコールをPCI ホストが実行するインタフェースからなる。プロセッサ機能は、前述のBAR2 空間上の制御レジスタにより、PCI ホストからの制御情報 (h_c, h_p, h_s, h_param, h_control) を受取り、また、PCI ホストに対してプロセッサの状態 (c_stat, a_stat, f_stat, s_stat, p_swi, p_param, その他コンソール入出力用レジスタ) を通知する。図 4.1 に BAR2 の詳細を示す。なお、PCI ホストが pe0_rstcount に非ゼロの 16 ビット値を書き込んだ場合、プロセッサ機能に至る PE0_RESET 信号がアサートされ、pe0_rstcount がデクリメントされて 0 に達した後に PE0_RESET 信号が解除される。

プロセッサ R/W の列は、プロセッサ機能側から許されるアクセス種類、PCI ホスト R/W の列は、PCI ホスト側から許されるアクセス種類を示す。このうち、c_stat は、h_c に対して非ゼロが書き込まれた場合、同じ値がセットされる。現時点では、セットされる値はコンソール入力の存在を示す URXINT (0x4) のみである。また、h_p に対して ARM 制御コマンド (0x000 から 0x7ff までのいずれか) が書き込まれた場合、a_stat に 1 (STATUS_BUSY) がセットされ、その後、各機能毎に規定された値がセットされる。同様に、h_p に対して FRV 制御コマンド (0x800 から 0xfff までのいずれか) が書き込まれた場合、f_stat に 1 (STATUS_BUSY) がセットされ、その後、各機能毎に規定された値がセットされる。h_s に対してハードウェアモニタ制御コマンドが書き込まれた場合、s_stat に 1 (STATUS_BUSY) がセットされ、その後、

各機能毎に規定された値がセットされる。

プロセッサ機能は、PCIからの h_c, h_p または h_s への書き込みに伴ってアサートされる前述の C_XINT 信号により、h_c, h_p または h_s の更新を検知し、BAR2 空間を直接参照することにより、h_c, h_p, h_s, h_param, h_control を受け取る。なお、現時点では、h_control の有効ビットは USE_SVC_INTR (ビット 0) および SINGLE_CYCLE (ビット 1) である。USE_SVC_INTR が 0 の場合、SWI 命令を検知したプロセッサ機能は、SWI 命令が PCI ホストにより実行されるよう、システムコールインタフェース (ARM の場合は a_stat=4 (STATUS_SWI) および p_swi, FRV の場合は f_stat=4 (STATUS_SWI)) により PCI ホストに対して SWI 情報を通知し、前述の C_XINT 信号により PCI ホストによる SWI 命令の実行完了を待ち合わせる。SINGLE_CYCLE が 1 の場合、1 サイクル動作後に、s_stat=2 (SCAN_SINGLE_STOP) または s_stat=3 (SCAN_ERROR_STOP) を通知する。ただし、SSRAM のバースト転送を行う外部キャッシュインタフェースに関しては、1 サイクルで停止する保証はなく、引き続き 1 サイクル動作が論理的に矛盾しない範囲内において外部キャッシュインタフェースが複数サイクル動作後に停止状態に遷移することがある。

4.1 コンソール入出力インタフェース

本インタフェースは、OS に必要なコンソール入出力を PCI ホストに接続するための論理インタフェースである。PCI ホストからのコンソール入力、次章にて説明するノンキャッシュブル主記憶空間の console_RXBUF1 または RXBUF2 に格納され、先頭アドレス、受信バッファ残量、受信可能状態が cons_rpr, cons_rcr, csUENDRX に各々設定される。その後、PCI ホストにより URXINT がアサートされ、OS に外部割り込みが通知される。同様に、OS からのコンソール出力は、console-TXBUF に格納され、先頭アドレス、長さ、未出力有状態が cons_txb, cons_txl, csTXEMPT に各々設定される。PCI ホストは csTXEMPT を監視し、未出力がある場合は画面に文字を出力する。関連レジスタは、以下の通り。

h_c	コンソール入力をプロセッサに通知する。現時点では URXINT (0x4) のみ
c_stat	PCI ホストが h_c へ書き込んだ場合、同一値が反映される。io_intsr の URXINT ビットがリセットされた場合は、同時に 0 リセットされる。
csUENDRX	コンソール入力検出時、console_rcr が 0 でなければ console_csr の受信終了フラグ US_ENDRX ビットを 1 とし、console_rcr が 0 の場合 0 とする。
cons_rcr	受信可能文字数 (PCI ホストにより入力文字数だけデクリメントされる)
cons_rpr	受信バッファ先頭アドレス (PCI ホストにより入力文字数だけインクリメントされる)
csTXEMPT	コンソール出力検出時、console_csr の送信文字なしフラグ US_TXEMPTY ビットを 0 とし、cons_txl が 0 になるまで送信し終えた場合 1 とする。
cons_txl	送信文字数 (PCI ホストにより出力文字数だけデクリメントされる)
cons_txb	送信バッファ先頭アドレス (PCI ホストにより出力文字数だけインクリメントされる)

4.2 プロセッサ制御インタフェース

本インタフェースは、PCI ホストによる、プロセッサの起動/停止、および、システムコール実行時のアーキテクチャレジスタの参照/更新のための論理インタフェースである。

PROC_NOP

何もしない。a_stat/f_stat は変化しない。

h_p	0x000
h_param	未使用

PROC_ARM_CONTINUE/PROC_FRV_CONTINUE

プロセッサは停止状態から走行状態に遷移し、プロセッサ内部に保持されている実行開始アドレスから命令の実行を開始する。プロセッサが走行状態である時に本機能を指定した場合、実行結果は予測不能である。本機能の実行完了時に、a_stat または f_stat が 1 から 0 (STATUS_NORMAL) に遷移する。

h_p	0x001/0x801
h_param	未使用

PROC_ARM_RESTART/PROC_FRV_RESTART

プロセッサは停止状態から走行状態に遷移し、h_param により指定される実行開始アドレスから命令の実行を開始する。プロセッサが走行状態である時に本機能を指定した場合、実行結果は予測不能である。本機能の実行完了時に、a_stat または f_stat が 1 から 0 (STATUS_NORMAL) に遷移する。

h_p	0x002/0x802
h_param	実行開始アドレス

PROC_ARM_FLUSH_STOP/PROC_FRV_FLUSH_STOP

本機能は、通常、システムリセット直後に発行される。プロセッサは、内蔵キャッシュのダーティラインを外部キャッシュに書き出した後、全てのキャッシュラインを無効化し、走行状態から停止状態に遷移する。外部キャッシュ機能を PCI ホスト上のソフトウェアとして実装している場合、Flush_Stop 動作中に、L2_ctl を通じて PCI ホストに対して外部キャッシュへの追い出し要求が通知されることがある。本機能の実行中、a_stat または f_stat が 1 から 2 (STATUS_FLUSH) に遷移し、実行完了時に 3 (STATUS_FLUSH_END) に遷移する。

h_p	0x003/0x803
h_param	未使用

PROC_ARM_STOP/PROC_FRV_STOP

プロセッサは、内蔵キャッシュの状態を保持したままで、走行状態から停止状態に遷移する。本機能の実行完了時に、a_stat または f_stat が 1 から 5 (STATUS_STOP) に遷移する。

h_p	0x004/0x804
h_param	未使用

Alter-REG

h_p で指定したプロセッサ内部のレジスタに値を書き込む。ただし ARM アーキテクチャレジスタ 15 番（プログラムカウンタ）への書き込みは無視する。実行開始アドレスの指定は、前述したプロセッサ制御コマンドを使用しなければならない。本コマンドは、a_stat または f_stat に STATUS_STOP または STATUS_FLUSH_STOP または STATUS_SWI が表示されているときのみ受付可能である。本コマンド受付後、a_stat または f_stat は STATUS_BUSY に遷移する。本コマンドの実行完了後、a_stat または f_stat は本コマンドを受け付けた時点の値に遷移する。

h_p	0x10X:usr レジスタ#0-15 0x12X:svc レジスタ#13-14 0x13X:abt レジスタ#13-14 0x14X:und レジスタ#13-14 0x15X:irq レジスタ#13-14 0x16X:fiq レジスタ#8-14 0x17X:implicit レジスタ#0-5 0x180:sftc(bit31-28のみ有効) 0x181:cpsr(bit31-28(NZVC) bit6-5(FT) bit4-0(M4-0)) 0x182:spsr_svc(bit31-28(NZVC) bit6-5(FT) bit4-0(M4-0)) 0x183:spsr_abt(bit31-28(NZVC) bit6-5(FT) bit4-0(M4-0)) 0x184:spsr_und(bit31-28(NZVC) bit6-5(FT) bit4-0(M4-0)) 0x185:spsr_irq(bit31-28(NZVC) bit6-5(FT) bit4-0(M4-0)) 0x186:spsr_fiq(bit31-28(NZVC) bit6-5(FT) bit4-0(M4-0)) 0x190:io_intmr 0x191:io_intsr 0x9aX:FRV アーキテクチャレジスタ#0-15 0x9bX:FRV アーキテクチャレジスタ#16-31 0x9c0:FRV Link Register 0x9c1:FRV ICC0 0x9c2:FRV ICC1
h_param	書き込み値

Display-REG

h_p で指定したプロセッサ内部のレジスタから p_param へ値を読み出す。ただし ARM アーキテクチャレジスタ 15 番からの読み出しは、ソースレジスタとしての読み出しではなく、プロセッサ内部に保持されている実行開始アドレス¹を読み出す。本コマンドは、a_stat および f_stat に STATUS_STOP または STATUS_FLUSH_STOP または STATUS_SWI が表示されているときのみ受付可能である。本コマンド受付後、a_stat または f_stat は STATUS_BUSY に遷移する。本コマンドの実行完了後、a_stat または f_stat は本コマンドを受け付けた時点の値に遷移する。

¹未実装

h_p	0x20X:usr レジスタ#0-15 0x22X:svc レジスタ#13-14 0x23X:abt レジスタ#13-14 0x24X:und レジスタ#13-14 0x25X:irq レジスタ#13-14 0x26X:fiq レジスタ#8-14 0x27X:implicit レジスタ#0-5 0x280:sftc(bit31-28のみ有効) 0x281:cpsr(bit31-28(NZVC) bit6-5(FT) bit4-0(M4-0)) 0x282:spsr_svc(bit31-28(NZVC) bit6-5(FT) bit4-0(M4-0)) 0x283:spsr_abt(bit31-28(NZVC) bit6-5(FT) bit4-0(M4-0)) 0x284:spsr_und(bit31-28(NZVC) bit6-5(FT) bit4-0(M4-0)) 0x285:spsr_irq(bit31-28(NZVC) bit6-5(FT) bit4-0(M4-0)) 0x286:spsr_fiq(bit31-28(NZVC) bit6-5(FT) bit4-0(M4-0)) 0x290:io_intmr 0x291:io_intsr 0xaaX:FRV アーキテクチャレジスタ#0-15 0xabX:FRV アーキテクチャレジスタ#16-31 0xac0:FRV Link Register 0xac1:FRV ICC0 0xac2:FRV ICC1
h_param	未使用

未定義機能

未定義機能を指定した場合、実行結果は予測不能である。

4.3 ハードウェアモニタ制御インタフェース

本インタフェースは、PCIホストがプロセッサ機能の内部状態を把握するための論理インタフェースである。PCIホストがSCAN_SINGLE_CYCLEをアサートした状態でプロセッサを起動すると、s_stat=2 (SCAN_SINGLE_STOP) または s_stat=3 (SCAN_ERROR_STOP) が通知され、プロセッサは停止状態となる。この状態において、以後、PCIホストが以下の機能を利用することができる。

SCAN_NOP

何もしない。s_statは変化しない。

h_s	0x0000
-----	--------

SCAN_STEP

s_stat=2 (SCAN_SINGLE_STOP) または s_stat=3 (SCAN_ERROR_STOP) が通知されている状態において、さらに1サイクル実行する。s_stat=1 (SCAN_BUSY) を経て、本機能の実行完了時に、s_stat=2 (SCAN_SINGLE_STOP) または s_stat=3 (SCAN_ERROR_STOP) が通知される。

h_s	0x0001
-----	--------

内部ラッチ書き込み

s_stat=2 (SCAN_SINGLE_STOP) または s_stat=3 (SCAN_ERROR_STOP) が通知されている状態において、内部状態にh_paramを書き込む。s_stat=1 (SCAN_BUSY) を経て、本機能の実行完了時に、

s_stat=2 (SCAN_SINGLE_STOP) または s_stat=3 (SCAN_ERROR_STOP) が通知される。内部状態は h_s により選択される。グループ内の詳細な状態割り当てについては別途規定する。

h_s	0x0100-0x01ff: SCAN_ARMIA1 グループ
h_s	0x0200-0x02ff: SCAN_ARMIF1 グループ
h_s	0x0300-0x03ff: SCAN_ARMDECODE グループ
h_s	0x0400-0x04ff: SCAN_HOSTDECODE グループ
h_s	0x0500-0x05ff: SCAN_FRVIA1 グループ
h_s	0x0600-0x06ff: SCAN_FRVIF1 グループ
h_s	0x0700-0x07ff: SCAN_FRVDECODE グループ
h_s	0x1000-0x17ff: SCAN_MAP グループ
h_s	0x1800-0x1fff: SCAN_SEL_READ グループ
h_s	0x2000-0x20ff: SCAN_EXEC_BRC グループ
h_s	0x2100-0x21ff: SCAN_EXEC_SFM グループ
h_s	0x2200-0x22ff: SCAN_EXEC_ALU グループ
h_s	0x2300-0x23ff: SCAN_EXEC_EAG グループ
h_s	0x2400-0x24ff: SCAN_EXEC_ME1 グループ
h_s	0x2500-0x25ff: SCAN_EXEC_ME2 グループ
h_s	0x2600-0x26ff: SCAN_EXEC_ME3 グループ
h_s	0x2700-0x27ff: SCAN_EXEC_ME4 グループ
h_s	0x2800-0x28ff: SCAN_EXEC_OP1 グループ
h_s	0x2900-0x29ff: SCAN_L2CT グループ
h_s	0x3000-0x37ff: SCAN_ARMIFCACHE_TAG グループ
h_s	0x3800-0x3fff: SCAN_FRVIFCACHE_TAG グループ
h_s	0x4000-0x4fff: SCAN_ARMIFCACHE_DAT グループ
h_s	0x5000-0x5fff: SCAN_FRVIFCACHE_DAT グループ
h_s	0x6000-0x7fff: SCAN_OPCACHE グループ
h_param	書き込み値

内部ラッチ読み出し

s_stat=2 (SCAN_SINGLE_STOP) または s_stat=3 (SCAN_ERROR_STOP) が通知されている状態において、内部状態を p_param に読み出す。s_stat=1 (SCAN_BUSY) を経て、本機能の実行完了時に、s_stat=2 (SCAN_SINGLE_STOP) または s_stat=3 (SCAN_ERROR_STOP) が通知される。内部状態は h_s により選択される。グループ内の詳細な状態割り当てについては別途規定する。

h_s	0x8000 0x0100-0x01ff: SCAN_ARMIA1 グループ
h_s	0x8000 0x0200-0x02ff: SCAN_ARMIF1 グループ
h_s	0x8000 0x0300-0x03ff: SCAN_ARMDECODE グループ
h_s	0x8000 0x0400-0x04ff: SCAN_HOSTDECODE グループ
h_s	0x8000 0x0500-0x05ff: SCAN_FRVIA1 グループ
h_s	0x8000 0x0600-0x06ff: SCAN_FRVIF1 グループ
h_s	0x8000 0x0700-0x07ff: SCAN_FRVDECODE グループ
h_s	0x8000 0x1000-0x17ff: SCAN_MAP グループ
h_s	0x8000 0x1800-0x1fff: SCAN_SEL_READ グループ
h_s	0x8000 0x2000-0x20ff: SCAN_EXEC_BRC グループ
h_s	0x8000 0x2100-0x21ff: SCAN_EXEC_SFM グループ
h_s	0x8000 0x2200-0x22ff: SCAN_EXEC_ALU グループ
h_s	0x8000 0x2300-0x23ff: SCAN_EXEC_EAG グループ
h_s	0x8000 0x2400-0x24ff: SCAN_EXEC_ME1 グループ
h_s	0x8000 0x2500-0x25ff: SCAN_EXEC_ME2 グループ
h_s	0x8000 0x2600-0x26ff: SCAN_EXEC_ME3 グループ
h_s	0x8000 0x2700-0x27ff: SCAN_EXEC_ME4 グループ
h_s	0x8000 0x2800-0x28ff: SCAN_EXEC_OP1 グループ
h_s	0x8000 0x2900-0x29ff: SCAN_L2CT グループ
h_s	0x8000 0x3000-0x37ff: SCAN_ARMIFCACHE_TAG グループ
h_s	0x8000 0x3800-0x3fff: SCAN_FRVIFCACHE_TAG グループ
h_s	0x8000 0x4000-0x4fff: SCAN_ARMIFCACHE_DAT グループ
h_s	0x8000 0x5000-0x5fff: SCAN_FRVIFCACHE_DAT グループ
h_s	0x8000 0x6000-0x7fff: SCAN_OPCACHE グループ
h_param	未使用

4.4 システムコールインタフェース

PCI ホストインタフェースにより、PCI ホストがプロセッサ機能からシステムコール要求を受けた場合、PCI ホストは、p_swi により指定されたシステムコールを実行しなければならない。システムコール検出時には、プロセッサ内部の一次キャッシュは無効化されているものの、外部キャッシュにはダーティラインが残っている可能性がある。このため、PCI ホストは、システムコールを実行する前に、ダーティラインを主記憶に追い出し、外部キャッシュを全て無効化しなければならない。もちろん、システムコール実行による主記憶内容更新の影響範囲を把握できる場合には、該当範囲のキャッシュラインのみを無効化すればよい。また、Exit を除くシステムコールについては、戻り値を Alter-REG により汎用レジスタ#0 に書き込んだ後、Continue によりプロセッサを再スタート (Longjmp の場合は Restart) しなければならない。

Exit

プロセッサがプログラムにより指定された停止命令を実行したことを示す。引続きプログラムを実行するためには、PCI ホストはプロセッサに対し、Flush_Stop および Restart を発行して内部状態をリセットしなければならない。本手順を行わずに Continue 等により実行を再開した場合、実行結果は予測不能である。

p_swi	1
汎用レジスタ#0-3	未使用

Read

入力装置から主記憶へのデータ転送要求である。主記憶への書き込みを伴うため、実行前に、影響を受ける範囲の外部キャッシュのダーティラインについては追い出しおよび無効化が必要である。

p_swi	3
汎用レジスタ#0	ファイルディスクリプタ
汎用レジスタ#1	転送先先頭物理アドレス
汎用レジスタ#2	バイト数
汎用レジスタ#3	未使用

Write

主記憶から出力装置へのデータ転送要求である。主記憶からの読み出しを伴うため、実行前に、参照される範囲の外部キャッシュのダーティラインについては追い出しが必要である。

p_swi	4
汎用レジスタ#0	ファイルディスクリプタ
汎用レジスタ#1	転送元先頭物理アドレス
汎用レジスタ#2	バイト数
汎用レジスタ#3	未使用

Open

ファイルオープン要求である。

p_swi	5
汎用レジスタ#0	ファイル名先頭物理アドレス
汎用レジスタ#1-2	第2, 第3 引数
汎用レジスタ#3	未使用

Close

ファイルクローズ要求である。

p_swi	6
汎用レジスタ#0	ファイルディスクリプタ
汎用レジスタ#1-3	未使用

Fstat

ファイル状態の取得要求である。

p_swi	7
汎用レジスタ#0	ファイルディスクリプタ
汎用レジスタ#1	格納先先頭物理アドレス
汎用レジスタ#2-3	未使用

Stat

ファイル状態の取得要求である。

p_swi	8
汎用レジスタ#0	ファイル名先頭物理アドレス
汎用レジスタ#1	格納先先頭物理アドレス
汎用レジスタ#2-3	未使用

Access

ファイルのアクセシビリティ検査要求である。

p_swi	9
汎用レジスタ#0	ファイル名先頭物理アドレス
汎用レジスタ#1	ファイルモード
汎用レジスタ#2-3	未使用

Fopen

ファイルオープン要求である。

p_swi	10
汎用レジスタ#0	ファイル名先頭物理アドレス
汎用レジスタ#1	ファイルモード指定文字列先頭物理アドレス
汎用レジスタ#2-3	未使用

Freopen

ファイル再オープン要求である。

p_swi	32
汎用レジスタ#0	ファイル名先頭物理アドレス
汎用レジスタ#1	ファイルモード指定文字列先頭物理アドレス
汎用レジスタ#2	オープン済ファイルポインタ 0xf0 ⇒ stdin, 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#3	未使用

Fread

ファイル読み込み要求である。主記憶への書き込みを伴うため、実行前に、影響を受ける範囲の外部キャッシュのダーティラインについては追い出しおよび無効化が必要である。

p_swi	11
汎用レジスタ#0	格納先先頭物理アドレス
汎用レジスタ#1	ブロックサイズ
汎用レジスタ#2	ブロック数
汎用レジスタ#3	ファイルポインタ 0xf0 ⇒ stdin に読み替え その他⇒無変更

Fgets

ファイル読み込み要求である。主記憶への書き込みを伴うため、実行前に、影響を受ける範囲の外部キャッシュのダーティラインについては追い出しおよび無効化が必要である。また、戻り値が0以外の場合、PCI ホストにおける fgets() の戻り値をプロセッサの主記憶物理アドレスに変換した値を汎用レジスタ#0に格納しなければならない。

p_swi	12
汎用レジスタ#0	格納先先頭物理アドレス
汎用レジスタ#1	バイト数
汎用レジスタ#2	ファイルポインタ 0xf0 ⇒ stdin に読み替え その他⇒無変更
汎用レジスタ#3	未使用

Fgetc

ファイル読み込み要求である。

p_swi	13
汎用レジスタ#0	ファイルポインタ 0xf0 ⇒ stdin に読み替え その他⇒無変更
汎用レジスタ#1-3	未使用

Ungetc

ファイル読み込み取消要求である。

p_swi	14
汎用レジスタ#0	取消文字
汎用レジスタ#1	ファイルポインタ 0xf0 ⇒ stdin に読み替え その他⇒無変更
汎用レジスタ#2-3	未使用

Getchar

標準入力読み込み要求である。

p_swi	15
汎用レジスタ#0-3	未使用

Fwrite

ファイル書き込み要求である。主記憶からの読み出しを伴うため、実行前に、参照される範囲の外部キャッシュのダーティラインについては追い出しが必要である。

p_swi	16
汎用レジスタ#0	先頭物理アドレス
汎用レジスタ#1	ブロックサイズ
汎用レジスタ#2	ブロック数
汎用レジスタ#3	ファイルポインタ 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更

Fputs

ファイル書き込み要求である。主記憶からの読み出しを伴うため、実行前に、参照される範囲の外部キャッシュのダーティラインについては追い出しが必要である。

p_swi	17
汎用レジスタ#0	先頭物理アドレス
汎用レジスタ#1	ファイルポインタ 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#2-3	未使用

Puts

標準出力書き込み要求である。主記憶からの読み出しを伴うため、実行前に、参照される範囲の外部キャッシュのダーティラインについては追い出しが必要である。

p_swi	31
汎用レジスタ#0	先頭物理アドレス
汎用レジスタ#1-3	未使用

Fputc

ファイル書き込み要求である。

p_swi	18
汎用レジスタ#0	出力文字
汎用レジスタ#1	ファイルポインタ 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#2-3	未使用

Putchar

標準出力書き込み要求である。

p_swi	19
汎用レジスタ#0	出力文字
汎用レジスタ#1-3	未使用

Fflush

ファイルフラッシュ要求である。

p_swi	20
汎用レジスタ#0	ファイルポインタ 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#1-3	未使用

Fseek

ファイルシーク要求である。

p_swi	21
汎用レジスタ#0	ファイルポインタ 0xf0 ⇒ stdin, 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#1	オフセット
汎用レジスタ#2	ポジション指定
汎用レジスタ#3	未使用

Ftell

ファイルポジション通知要求である。

p_swi	22
汎用レジスタ#0	ファイルポインタ 0xf0 ⇒ stdin, 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#1-3	未使用

Fgetpos

ファイルポジション通知要求である。

p_swi 汎用レジスタ#0	39 ファイルポインタ 0xf0 ⇒ stdin, 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#1	格納先主記憶物理アドレス
汎用レジスタ#2-3	未使用

Rewind

ファイルリワインド要求である。

p_swi 汎用レジスタ#0	33 ファイルポインタ 0xf0 ⇒ stdin, 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#1-3	未使用

Fclose

ファイルクローズ要求である。

p_swi 汎用レジスタ#0	23 ファイルポインタ 0xf0 ⇒ stdin, 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#1-3	未使用

Clearerr

ファイルエラー状態消去要求である。

p_swi 汎用レジスタ#0	24 ファイルポインタ 0xf0 ⇒ stdin, 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#1-3	未使用

Ferror

ファイルエラー状態通知要求である。

p_swi 汎用レジスタ#0	25 ファイルポインタ 0xf0 ⇒ stdin, 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#1-3	未使用

Fileno

ファイルディスクリプタ通知要求である。

p_swi 汎用レジスタ#0	26 ファイルポインタ 0xf0 ⇒ stdin, 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#1-3	未使用

Feof

ファイル終端検査要求である。

p_swi 汎用レジスタ#0	28 ファイルポインタ 0xf0 ⇒ stdin, 0xf1 ⇒ stdout, 0xf2 ⇒ stderr に各々読み替え その他⇒無変更
汎用レジスタ#1-3	未使用

Unlink

ファイル削除要求である。

p_swi 汎用レジスタ#0	27 ファイル名先頭物理アドレス
汎用レジスタ#1-3	未使用

Remove

ファイル削除要求である。

p_swi 汎用レジスタ#0	38 ファイル名先頭物理アドレス
汎用レジスタ#1-3	未使用

Rename

ファイル改名要求である。

p_swi 汎用レジスタ#0	29 旧ファイル名先頭物理アドレス
汎用レジスタ#1	新ファイル名先頭物理アドレス
汎用レジスタ#2-3	未使用

Time

時刻取得要求である。汎用レジスタ#0 が 0 の場合、格納先物理アドレスも 0 である。

p_swi 汎用レジスタ#0	30 格納先先頭物理アドレス
汎用レジスタ#1-3	未使用

Mktemp

一時ファイル作成要求である。返り値が 0 以外の場合、PCI ホストにおける mktemp() の返り値をプロセッサの主記憶物理アドレスに変換した値をレジスタ#0 に格納しなければならない。

p_swi 汎用レジスタ#0	34 ファイル名テンプレート先頭物理アドレス
汎用レジスタ#1-3	未使用

Tmpfile

ファイルポインタ通知要求である。

p_swi	35
汎用レジスタ#0-3	未使用

Isatty

ファイルディスクリプタ検査要求である。

p_swi	36
汎用レジスタ#0	ファイルディスクリプタ
汎用レジスタ#1-3	未使用

Ttyname

デバイス名通知要求である。戻り値が0以外の場合、PCI ホストにおける ttyname() の戻り値をプロセスの主記憶物理アドレスに変換した値をレジスタ#0に格納しなければならない。

p_swi	37
汎用レジスタ#0	ファイルディスクリプタ
汎用レジスタ#1-3	未使用

Setjmp

現コンテキストの記録要求である。PCI ホストは、先頭アドレスから順に、レジスタ#1から#14 (Setjmp を呼び出した命令の次アドレス) までを格納しなければならない。レジスタ#0には0を格納する。主記憶への書き込みを伴うため、実行前に、影響を受ける範囲の外部キャッシュのダーティラインについては追い出しおよび無効化が必要である。

p_swi	40
汎用レジスタ#0	格納先先頭物理アドレス
汎用レジスタ#1-3	未使用

Longjmp

指定コンテキストの復元要求である。PCI ホストは、先頭アドレスから順に、レジスタ#1から#13、および、#15を復元し、Setjmp を呼び出したコンテキストに戻らなければならない。ただし、#15への復元には、Alter-REGではなくRestartを使用しなければならない。レジスタ#0には、汎用レジスタ#1により指定された値 (Setjmp と区別するために通常は0以外) を格納する。主記憶からの読み出しを伴うため、実行前に、参照される範囲の外部キャッシュのダーティラインについては追い出しが必要である。

p_swi	41
汎用レジスタ#0	格納先先頭物理アドレス
汎用レジスタ#1	戻り値
汎用レジスタ#2-3	未使用

未定義

p_swi が未定義の場合は、ハードウェア異常である。

Chapter 5

実装すべき命令および機能

uClinux を走行するため、および、OS 不在の環境において評価用ベンチマークプログラムを単独走行するために、OROCHI システムが最低限備えなければならない機能を示す。大まかに、ARM/FPGA モデルでは後述する ARM 命令をシングルスレッド実行、OROCHI/FPGA モデルおよび OROCHI/専用 LSI モデルでは後述する ARM 命令および VLIW 命令をマルチスレッド実行できなければならない。

5.1 ARM 命令

ARM Architecture Reference Manual ARM DDI 0100E に規定される機能のうち、Thumb および DSP を備えない ARMv4 レベルを満たす必要がある。表 5.1 および表 5.2 に実行すべき ARM 命令を示す。本命令は、gcc-4.1.1 および binutils-2.17 を用いて、gcc -march=armv4 -msoft-float -O2 により生成されるアプリケーションプログラムを実行するために最低限必要な一般命令 (ADC, ADD, AND, B, BL, BIC, CMN, CMP, EOR, LDM(1), LDR, LDRB, LDRH, LDRSB, LDRSH, MLA, MOV, MUL, MVN, ORR, RSB, RSC, SBC, SMULL, STM(1), STR, STRB, STRH, SUB, SWI, TEQ, TST, UMLAL, UMULL) に、uClinux を動作させるために必要な命令 (BX, LDM(2), LDM(3), LDRBT, LDRT, MSR, MRS, STM(2), STRBT, STRT) を加えたものである。

5.2 ARM コアに必要な機能

上記命令実行機能の他に実装すべき機能は以下の通りである。

- CPSR, FIQ, IRQ, Supervisor, Abort, Undefined, User, System の各モード、および、各モード用 SPSR/汎用レジスタ
- モード切替えによる、SWI 命令のホスト実行機能および SVC 割り込み生成機能
- I/O 割り込み (intsr.bit6:TC20I, intsr.bit5:TC10I, intsr.bit2:URXINT, intsr.bit1:UTXINT)

5.3 内部命令

OROCHI システムが前述の ARM 命令を分解実行するために装備すべき内部命令を表 5.3 に示す。

5.4 VLIW 命令

OROCHI システムが直接実行しなければならない VLIW 命令を表 5.4 に示す。なお、本 VLIW 命令は内部命令の一部を含む。なお、後述するように、レジスタ空間は VLIW コア専用設ける。

- 実行すべき命令は、ADD, SUB, AND, OR, XOR, NOT, SLL, SRL, SRA, LD, ST, LDF, STF, MOVGS, MOVSG, SETHI, SETLO, SETLOS, Bicc, BicclR, CALLL, JMPL, CALL, TIicc

[全体]																			
8		32		32		35						32			5 (48+96=144)				
2	5	1	32	32	6	4	4	4	2	2	5	2	4	2	32	2	1	2	
v	patt	pred	pc	target	ghr	cond	type	opcd	dct	drt	dcm	s1t	s1n	s2t	s2n	sst	ssn	sct	
分解時 フラグ	命令 の型	分岐予測	PC値	分岐先 予測値	分岐予測	0000:EQ 0001:NE 0010:CS 0011:CC 0100:HI 0101:FL 0110:VS 0111:VC 1000:HI 1001:LS 1010:GE 1011:LT 1100:GT 1101:LE 1110:AL 1111:NV			00:nop	00:nop	----- -----	00:nop 01:PC 10:ARM 11:SC 更新	----- -----	00:nop 01:imm 10:ARM 11:SC 更新	----- imm値 REG.No REG.No REG.No REG.No	00:nop 01:imm 10:CC 11:SC	----- imm値	----- 10:ARM CC	

[命令タイプ別]																			
8		32		32		35						32			5 (48+96=144)				
2	5	1	32	32	6	4	4	4	2	2	5	2	4	2	32	2	1	2	
v	patt	pred	pc	target	ghr	cond	type	opcd	dct	drt	dcm	s1t	s1n	s2t	s2n	sst	ssn	sct	
		-----		-----	-----	-----	0000	NOP											
		-----	PC+8	-----	-----	CCND	0010	0000:LSL 0001:LSR 0010:ASR 0011:ROR 0100:RFX	00:nop 11:SC 更新	11:裏R	裏R.No	01:PC 10:ARM	----- REG.No	01:imm 10:ARM	imm REG.No	-----		CC	
		-----		-----	-----	CCND	0011	0000:OVL 0001:OCL 0010:OSH 0011:OCH 0100:ISL 0101:ICL 0110:1SH 0111:1CH 1000:2SL 1001:2CL 1010:2SH 1011:2CH 1100:3SL 1101:3CL 1110:3SH 1111:3CH		11:裏R	裏R.No	10:ARM 11:裏R	REG.No REG.No	10:ARM 11:裏R	REG.No REG.No	-----		CCND依	
		-----	PC+8	RASの imp先 予測値	-----	CCND	0100	0000:AND 0001:BCR 0010:SUB 0011:RSB 0100:ADD 0101:ADC 0110:SEC 0111:RSC 1000:STP 1001:TEQ 1010:CMP 1011:CMN 1100:ORR 1101:MOV 1110:BIC 1111:MVN	00:nop 10:CC 更新	00:nop 10:ARM 11:裏R	REG.No 裏R.No	00:nop 01:PC 10:ARM 11:裏R	s1 s1 s1 s1 s1 s1 s1 s1 s1 s1 s2 s2 s2 s2	00:nop 01:imm 10:ARM 11:裏R	s2 s2 s2 s2 s2 s2 s2 s2 s2 s2 s2 s2 s2 s2	00:nop 01:imm 10:CC 11:SC	SS SS SS SS SS SS SS SS SS SS SS SS SS SS	10:CC 10:CC 10:CC	
		-----		-----	-----	CCND	0110	0000:EAD 0001:RSB 0010:N>C 0100:A<C 0110:64C 1000:ABS 1010:N>C 1100:N<C 1110:CSL	00:nop 10:CC 更新 11:SC 更新	00:nop 10:ARM 11:裏R	REG.No 裏R.No	00:nop 10:ARM 11:裏R	s1	00:nop 01:imm 10:ARM 11:裏R	s2	00:nop 11:SC	-----		10:CC
		-----		RASの imp先 予測値	-----	CCND	1000	0000:LUB 0001:LUB 0010:LSB 0011:LSW 0100:LW 1000:MRS	10:LEM 3用	10:ARM	REG.No	10:ARM 11:裏R	Rn 裏R.No	-----	-----	-----	01:imm	R-bit	CCND依
		-----		-----	-----	CCND	1001	0000:SUB 0001:SUH 0100:SW 1000:MS0 1001:MS1 1010:MS8 1011:MS9	10:CC 00:nop #15 #15 #15 #15	00:nop	10:ARM 11:裏R	Rn 裏R.No	10:ARM 10:ARM	Rd imm Reg.No	01:imm imm Reg.No	01:imm	R-bit	CCND依	
		-----	PC+4	-----	-----	CCND	1010	SWI	-----	-----	-----	-----	-----	01:imm	SWI	-----	-----	CCND依	
		-----	PC+4	-----	-----	CCND	1011	undef	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	CCND依
		分岐予測	PC+8	target	分岐予測	CCND	1100	0000:BR 0001:BL	11:SC 予測HP?	BL時は ARMREG	BL時は #14	01:PC	-----	01:imm	NextPC (PC+4)	-----	分岐予測 方向	CC	

図 5.3: 実装すべき内部命令

5.5 VLIW コアに必要な機能

上記命令実行機能の他に実装すべき機能は以下の通りである。

- 32bit 汎用レジスタ 32 本, ICC0/1, LR
- 32bit メディアレジスタ 32 本

- 0x0f000000 : console-TXBUF
 - 0x0f001000 : console-RXBUF1
 - 0x0f002000 : console-RXBUF2
 - 0x0f004000 : FRV-SVC-CODE
 - 0x0f004010 : FRV-SVC-GR4
 - 0x0f004014 : FRV-SVC-GR5
 - 0x0f004018 : FRV-SVC-GR6
 - 0x0f00401C : FRV-SVC-GR7
 - 0x0f004020 : FRV-SVC-RETV
 - 0x0f004100 : 書き込み時, PCI ホストの cam_capt() を呼び出す
 - 0x0f004104 : 書き込み時, PCI ホストの x11_updt() を呼び出す
 - 0x0f100000 : CAM_L_BUF (ステレオ画像処理左目カメラ画像)
 - 0x0f180000 : CAM_R_BUF (ステレオ画像処理右目カメラ画像)
 - 0x0f200000 : CAM_W_BUF (ステレオ画像処理作業領域画像)
 - 0x0f280000 : CAM_D_BUF (ステレオ画像処理奥行き画像)
- 未実装空間 (0x10000000-0xfffffff)
 - I/O および制御レジスタ空間 (0xffff0000-0xffffffff), コンソール I/O 機能およびタイマ割り込み機能
 - 0xffff0000(R) : CPUID
 - 0xffff0100(W) : FRV_SWLCONTIUNE
 - 0xffff0104(W) : FRV_SWLSTOP
 - 0xffff0108(W) : FRV_PROC_RESTART
 - 0xffff100(R) : IVR
 - 0xffff108(R/W): CIVR/INTSR
 - 0xffff10C(R) : IPR
 - 0xffff110(R/W): IMR
 - 0xffff114(R) : INTSR
 - 0xffff120(R/W): IECR
 - 0xffff124(W) : IDCR
 - 0xffff128(W) : CLRINTSR
 - 0xffff130(W) : EOI
 - 0xfffe0040(W) : T1CCR
 - 0xfffe0050(R/W): T1CV
 - 0xfffe005C(R/W): T1RC
 - 0xfffe0060(R) : T1SR
 - 0xfffd0000(W) : console-CSRTX
 - 0xfffd0014(R) : console-CSR
 - 0xfffd001C(W) : console-THR
 - 0xfffd0030(W) : console-RPR
 - 0xfffd0034(R/W): console-RCR
 - 0xfffd0038(W) : console-TXB
 - 0xfffd003C(W) : console-TXL

Chapter 6

専用 LSI モデルにおけるピン配置

XC2V6000 とドータボードの物理インタフェースは、SSRAM インタフェース信号 (110 本)、および、プロセッサ制御インタフェース信号 (40 本) の合計 150 本からなる。さらに、スキャン用信号 4 本、テスト用信号 10 本、クロック、クリア、プリセットの 3 本を加えた合計 167 本が専用 LSI に最低限必要な I/O である。ドータボード上の HQFP240 および QFP208 の各 LSI ソケットには、以上の I/O が各々 1 組接続されており、XC2V6000 から独立に制御できる。GP600M-PCI 上のコネクタ CN1 および CN2 は HQFP240 用、CN3 および CN4 は QFP208 用信号を収容する。専用 LSI の I/O には、駆動力 2mA のドライバを推奨する。

6.1 HQFP240

専用 LSI (CE77 : HQFP240) のピン配置制約 (富士通による) は次の通り。コア電圧は 2.5V。I/O は 3.3V。電源ピンを除く合計 198 ピンが CN1 および CN2 を経由して XC2V6000 に接続される。一般信号用のうち 150 ピンを SSRAM インタフェースおよび PCI インタフェースとして使用する。

- VDDI(2.5V) : 計 12 ピン (25, 36, 42, 77, 89, 101, 145, 156, 162, 197, 209, 221)
- VDDE(3.3V) : 計 10 ピン (6, 18, 54, 66, 114, 126, 138, 174, 186, 234)
- VSS : 計 20 ピン (7, 19, 30, 43, 55, 67, 78, 90, 102, 115, 127, 139, 150, 163, 175, 187, 198, 210, 222, 235)
- JTAG 用 TDI, TDO, TMS, TCK, TRST : 計 5 ピン (未使用時はプルアップ)
- テストピン XSM, XTST, SMCK, MST, SAT : 計 5 ピン (未使用時はプルアップ)
- クロック, クリア, プリセット入力ピン : 計 3 ピン (互いに隣接配置しない。1つの電源グループ内に配置できる本信号は最大 2ヶ所。電源ピン (VSS か VDD) に近接配置。入力ピンでガードする等により出力ピンから離す。隣接する電源グループの範囲内に出力バッファや入力クロックバッファを配置しない。出力/双方向/入力クロックバッファと入力クロックバッファを隣接配置しない)
- 一般信号用 : 残り 185 ピン。同時スイッチングする出力ピンおよび双方向ピンは、電源ピンに近接配置。電源ピンごとに分散配置。電源グループに対する同時スイッチング出力バッファの許容本数を守る。同時スイッチングする出力バッファや、その近傍の出力バッファを他 LSI のクロック入力としな。また、ドライババリティの大きな出力ピンは中央に配置。

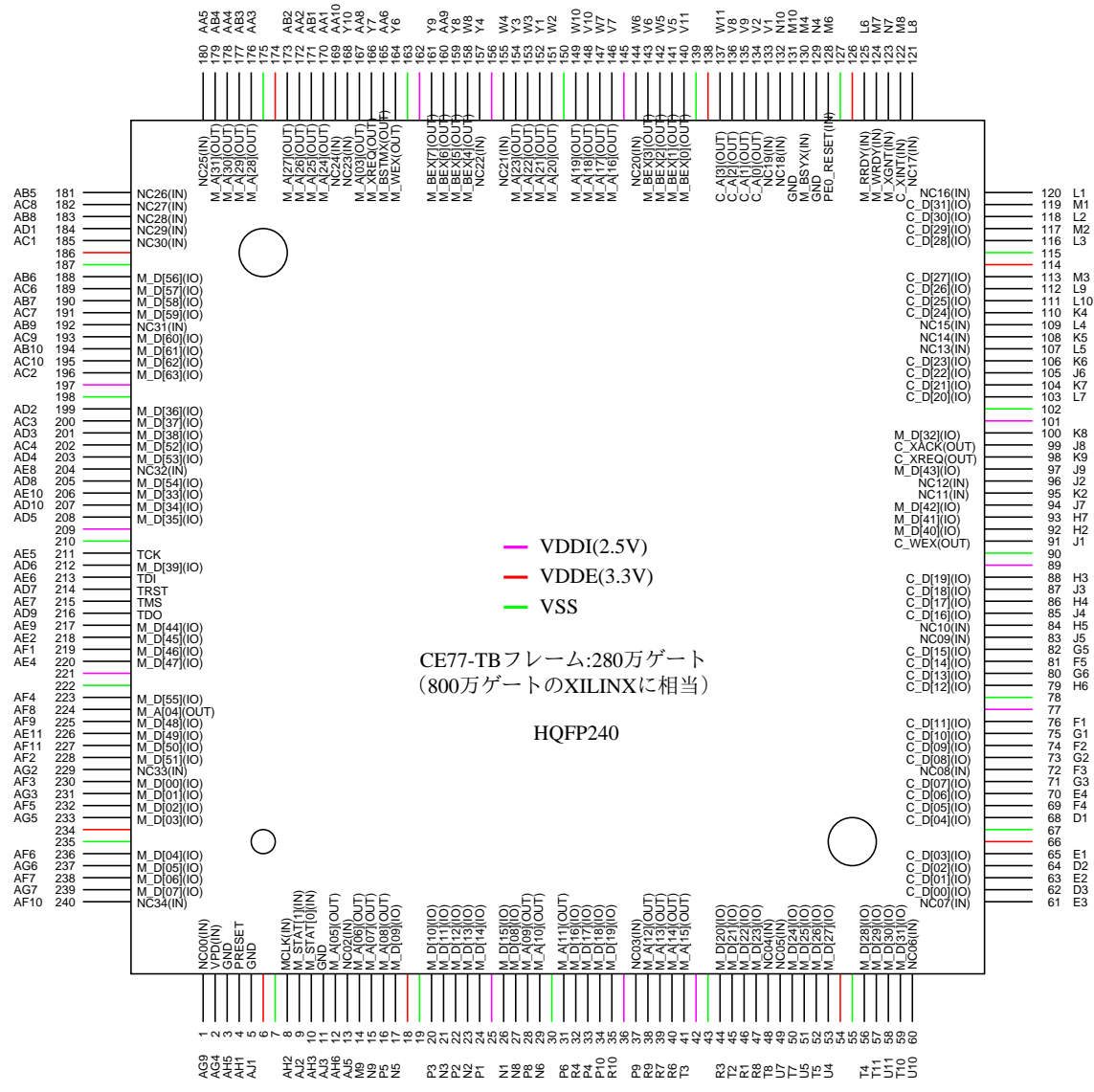


図 6.1: HQFP240 のピン配置 (周辺の記号は XILINX の対応ピン)

6.2 QFP208

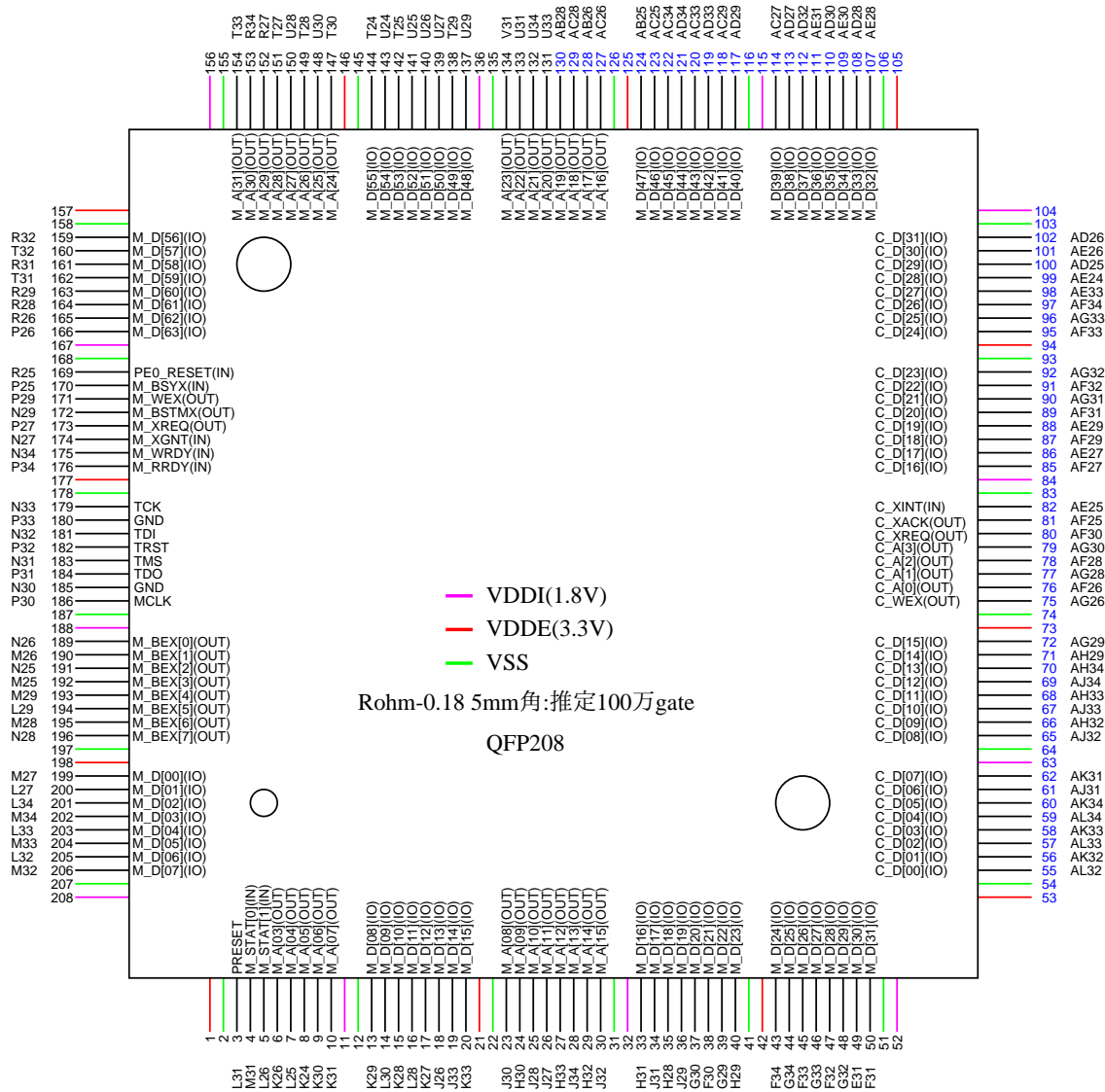


図 6.2: QFP208 のピン配置 (周辺の記号は XILINX の対応ピン)

専用 LSI (ROHM: QFP208) のピン配置制約 (VDEC による) は次の通り。コア電圧は 1.8V。I/O は 3.3V。電源ピンを除く合計 160 ピンが CN3 および CN4 を経由して XC2V6000 に接続される。一般信号用のうち 150 ピンを SSRAM インタフェースおよび PCI インタフェースとして使用する。

- VDD(1.8V) : 計 12 ピン (11, 32, 52, 63, 84, 104, 115, 136, 156, 167, 188, 208)
- VDDO(3.3V) : 計 12 ピン (1, 21, 42, 53, 73, 94, 105, 125, 146, 157, 177, 198)
- VSS : 計 24 ピン (2, 12, 22, 31, 41, 51, 54, 64, 74, 83, 93, 103, 106, 116, 126, 135, 145, 155, 158, 168, 178, 187, 197, 207)
- JTAG 用 TDI, TDO, TMS, TCK, TRST : 計 5 ピン (未使用時はプルアップ)
- クロック, クリア, プリセット入力ピン : 計 3 ピン (互いに隣接配置しない。1つの電源グループ内に配置できる本信号は最大 2ヶ所。電源ピン (VSS か VDD) に近接配置。入力ピンでガードする等により出力ピンから離す。隣接する電源グループの範囲内に出力バッファや入力クロックバッファを配置しない。出力/双方向/入力クロックバッファと入力クロックバッファを隣接配置しない)

- 一般信号用：残り 152 ピン。同時スイッチングする出力ピンおよび双方向ピンは、電源ピンに近接配置。電源ピンごとに分散配置。電源グループに対する同時スイッチング出力バッファの許容本数を守る。同時スイッチングする出力バッファや、その近傍の出力バッファを他 LSI のクロック入力としな
い。また、ドライバリティの大きな出力ピンは中央に配置。

I/O バッファには以下を推奨する

- PC30B01 入出力用。出力用 I ピンおよび OEN の容量は各 22fF (INV × 3 に相当)。OEN × 8 ピン
を駆動するには 8 倍 INV が必要。
- PC30D01 入力専用。PAD から CIN の遅延は負荷 FO4/FO32 で約 200ps/300ps。INV × 32 まで直
接駆動可能。
- PC30O01 出力専用。I から PAD の遅延は負荷 20pF/60pF で 5ns/10ns。

Chapter 7

HQFP240に関する提供情報

7.1 ブロック図, フロアプラン, クロック系統図

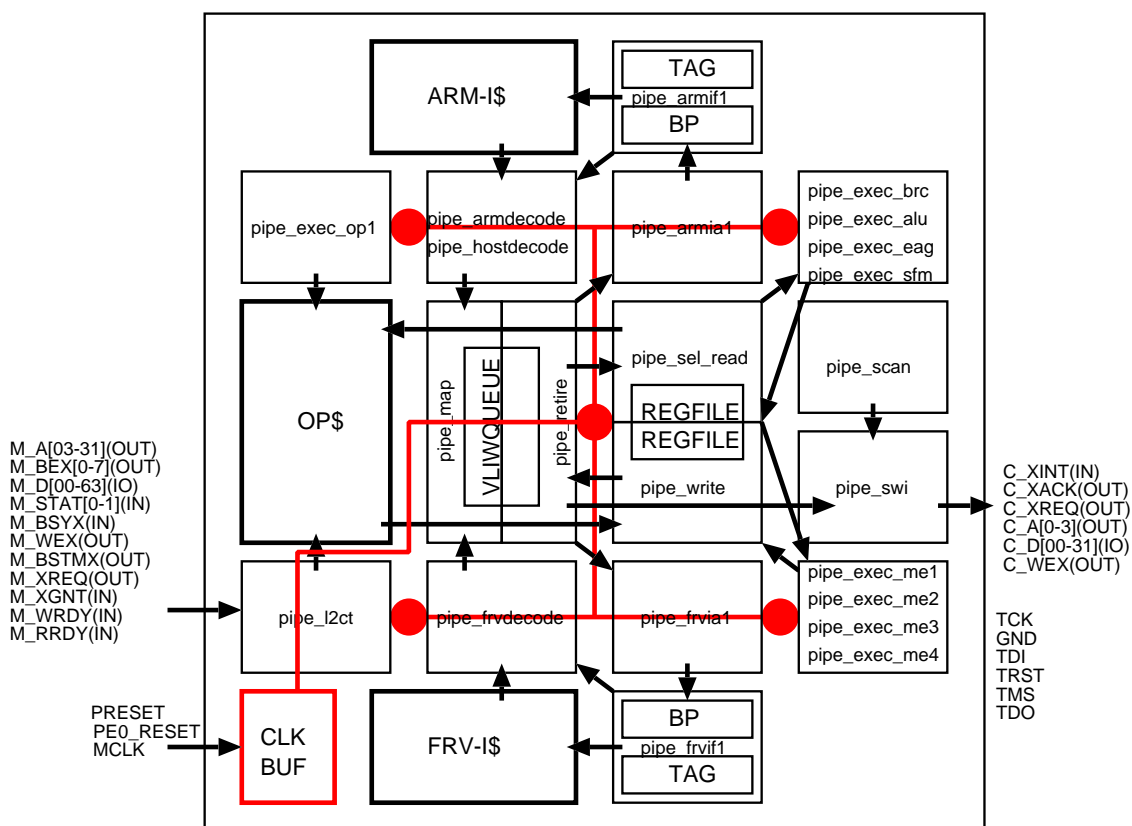


図 7.1: 専用 LSI のブロック図兼フロアプラン

表 7.1: クロック情報

クロックピン名	クロックグループ名	周期 [ps]	Duty 変動	ジッタ [ps]	位相 [ps]
MCLK	-	15151.5ps	45%-55%	1000	0
TCK	-	15151.5ps	45%-55%	1000	0

7.2 消費電力

表 7.2: モジュール消費電力

モジュール名	基準周波数	動作率	モジュール名	基準周波数	動作率
pipe_armia	66MHz	20%	pipe_exec_alu	66MHz	20%
pipe_armif	66MHz	20%	pipe_exec_eag	66MHz	20%
pipe_armdecode	66MHz	20%	pipe_exec_sfm	66MHz	20%
pipe_hostdecode	66MHz	20%	pipe_exec_me1	66MHz	20%
pipe_map	66MHz	20%	pipe_exec_me2	66MHz	20%
pipe_frvia	66MHz	20%	pipe_exec_me3	66MHz	20%
pipe_frivf	66MHz	20%	pipe_exec_me4	66MHz	20%
pipe_frdecode	66MHz	20%	pipe_exec_yaop1	66MHz	20%
pipe_sel_read	66MHz	20%	pipe_L2	66MHz	20%
pipe_exec_brc	66MHz	20%	pipe_swi	66MHz	20%

表 7.3: メモリ構成表

RAM インスタンス名	使用モジュール	メモリタイプ	bit 幅	エントリ数	容量
a1_phr.i1_phr	i1_phr	1RW/1R	4	4096	16384
f1_phr1.i1_phr	i1_phr	1RW/1R	4	4096	16384
a1_tag0.i1_tag0	i1_tag	1RW/1R	23	64	1472
a1_tag1.i1_tag0	i1_tag	1RW/1R	23	64	1472
a1_tag2.i1_tag0	i1_tag	1RW/1R	23	64	1472
a1_tag3.i1_tag0	i1_tag	1RW/1R	23	64	1472
a1_dat0.i1_data	i1_dat	1RW	512	64	32768
a1_dat1.i1_data	i1_dat	1RW	512	64	32768
a1_dat2.i1_data	i1_dat	1RW	512	64	32768
a1_dat3.i1_data	i1_dat	1RW	512	64	32768
way0.f1_tag	i1_tag	1RW/1R	23	64	1472
way1.f1_tag	i1_tag	1RW/1R	23	64	1472
way2.f1_tag	i1_tag	1RW/1R	23	64	1472
way3.f1_tag	i1_tag	1RW/1R	23	64	1472
way0.f1_dat_0	i1_dat	1RW	512	64	32768
way1.f1_dat_0	i1_dat	1RW	512	64	32768
way2.f1_dat_0	i1_dat	1RW	512	64	32768
way3.f1_dat_0	i1_dat	1RW	512	64	32768
tag_core_ce77_0	d1_tag_core_ce77	1RW	24	64 (128)	1536 (3072)
tag_core_ce77_1	d1_tag_core_ce77	1RW	24	64 (128)	1536 (3072)
tag_core_ce77_2	d1_tag_core_ce77	1RW	24	64 (128)	1536 (3072)
tag_core_ce77_3	d1_tag_core_ce77	1RW	24	64 (128)	1536 (3072)
dat_core_ce77_0	d1_dat_core_ce77	1RW	512	64 (128)	32768 (65536)
dat_core_ce77_1	d1_dat_core_ce77	1RW	512	64 (128)	32768 (65536)
dat_core_ce77_2	d1_dat_core_ce77	1RW	512	64 (128)	32768 (65536)
dat_core_ce77_3	d1_dat_core_ce77	1RW	512	64 (128)	32768 (65536)

表 7.4: RAM 消費電力

RAM インスタンス名	基準周波数	インヒビット動作率	R/W 動作割合
a1_phr.i1_phr (A port)	66MHz	20%	50%
a1_phr.i1_phr (B port)	66MHz	20%	-
f1_phr1.i1_phr (A port)	66MHz	20%	50%
f1_phr1.i1_phr (B port)	66MHz	20%	-
a1_tag0.i1_tag0 (A port)	66MHz	20%	50%
a1_tag0.i1_tag0 (B port)	66MHz	20%	-
a1_tag1.i1_tag0 (A port)	66MHz	20%	50%
a1_tag1.i1_tag0 (B port)	66MHz	20%	-
a1_tag2.i1_tag0 (A port)	66MHz	20%	50%
a1_tag2.i1_tag0 (B port)	66MHz	20%	-
a1_tag3.i1_tag0 (A port)	66MHz	20%	50%
a1_tag3.i1_tag0 (B port)	66MHz	20%	-
a1_dat0.i1_data	66MHz	20%	50%
a1_dat1.i1_data	66MHz	20%	50%
a1_dat2.i1_data	66MHz	20%	50%
a1_dat3.i1_data	66MHz	20%	50%
way0.f1_tag (A port)	66MHz	20%	50%
way0.f1_tag (B port)	66MHz	20%	-
way1.f1_tag (A port)	66MHz	20%	50%
way1.f1_tag (B port)	66MHz	20%	-
way2.f1_tag (A port)	66MHz	20%	50%
way2.f1_tag (B port)	66MHz	20%	-
way3.f1_tag (A port)	66MHz	20%	50%
way3.f1_tag (B port)	66MHz	20%	-
way0.f1_dat_0	66MHz	20%	50%
way1.f1_dat_0	66MHz	20%	50%
way2.f1_dat_0	66MHz	20%	50%
way3.f1_dat_0	66MHz	20%	50%
tag_core_ce77_0	66MHz	20%	50%
tag_core_ce77_1	66MHz	20%	50%
tag_core_ce77_2	66MHz	20%	50%
tag_core_ce77_3	66MHz	20%	50%
dat_core_ce77_0	66MHz	20%	50%
dat_core_ce77_1	66MHz	20%	50%
dat_core_ce77_2	66MHz	20%	50%
dat_core_ce77_3	66MHz	20%	50%

7.3 AC 特性

表 7.5: AC 特性

端子名	基準クロック	クロック極性	入力 setup/hold	出力 min/max	外部負荷容量
全て	MCLK	↑	5ns/5ns	3ns/10ns	20pF

7.4 端子情報

Low-Noise Type を使用. PU/PD 無し. 出力電流は 2mA. 外部負荷 (FPGA の入力容量) は 20pF. 動作周波数は 66MHz. 動作率は 5%. 双方向 SW 率は 0.5. OUTPUT(TB Low-Noise 2mA) には OT3LLR2X, INPUT(TB) には IT3HX, BIDIR(TB Low-Noise 2mA) には B3NNLLR2X を各々使用.

表 7.6: 端子情報

端子	名前	I/O	使用 buff	同時 sg	用途	端子	名前	I/O	使用 buff	同時 sw	用途
1	NC00	IN	IT3HX	-	open	121	NC17	IN	IT3HX	-	open
2	VPD	IN	IT3HX	-	貫通防止	122	C_XINT	IN	IT3HX	-	regint
3	GND	IN	IT3HX	-	-	123	M_XGNT	IN	IT3HX	-	memgmt
4	PRESET	IN	IT3HX	-	未使用	124	M_WRDY	IN	IT3HX	-	memwrdrdy
5	GND	IN	IT3HX	-	-	125	M_RRDY	IN	IT3HX	-	memrrdy
8	MCLK	IN	IT3KH	-	CLOCK	128	PEO_RESET	IN	IT3HX	-	RESET
9	M_STAT[1]	IN	IT3HX	-	-	129	GND	IN	IT3HX	-	-
10	M_STAT[0]	IN	IT3HX	-	memstat	130	M_BSYX	IN	IT3HX	-	busy
11	GND	IN	IT3HX	-	-	131	GND	IN	IT3HX	-	-
12	M_A[05]	OUT	OT3LLR2X	MA	-	132	NC18	IN	IT3HX	-	open
13	NC02	IN	IT3HX	-	open	133	NC19	IN	IT3HX	-	open
14	M_A[06]	OUT	OT3LLR2X	MA	-	134	C_A[0]	OUT	OT3LLR2X	CA	regaddr
15	M_A[07]	OUT	OT3LLR2X	MA	-	135	C_A[1]	OUT	OT3LLR2X	CA	-
16	M_A[08]	OUT	OT3LLR2X	MA	-	136	C_A[2]	OUT	OT3LLR2X	CA	-
17	M_D[09]	IO	B3NNLLR2X	MD	-	137	C_A[3]	OUT	OT3LLR2X	CA	-
20	M_D[10]	IO	B3NNLLR2X	MD	-	140	M_BEX[0]	OUT	OT3LLR2X	MB	memmask
21	M_D[11]	IO	B3NNLLR2X	MD	-	141	M_BEX[1]	OUT	OT3LLR2X	MB	-
22	M_D[12]	IO	B3NNLLR2X	MD	-	142	M_BEX[2]	OUT	OT3LLR2X	MB	-
23	M_D[13]	IO	B3NNLLR2X	MD	-	143	M_BEX[3]	OUT	OT3LLR2X	MB	-
24	M_D[14]	IO	B3NNLLR2X	MD	-	144	NC20	IN	IT3HX	-	open
26	M_D[15]	IO	B3NNLLR2X	MD	-	146	M_A[16]	OUT	OT3LLR2X	MA	-
27	M_D[08]	IO	B3NNLLR2X	MD	memdata	147	M_A[17]	OUT	OT3LLR2X	MA	-
28	M_A[09]	OUT	OT3LLR2X	MA	-	148	M_A[18]	OUT	OT3LLR2X	MA	-
29	M_A[10]	OUT	OT3LLR2X	MA	-	149	M_A[19]	OUT	OT3LLR2X	MA	-
31	M_A[11]	OUT	OT3LLR2X	MA	-	151	M_A[20]	OUT	OT3LLR2X	MA	-
32	M_D[16]	IO	B3NNLLR2X	MD	-	152	M_A[21]	OUT	OT3LLR2X	MA	-
33	M_D[17]	IO	B3NNLLR2X	MD	-	153	M_A[22]	OUT	OT3LLR2X	MA	-
34	M_D[18]	IO	B3NNLLR2X	MD	-	154	M_A[23]	OUT	OT3LLR2X	MA	-
35	M_D[19]	IO	B3NNLLR2X	MD	-	155	NC21	IN	IT3HX	-	open
37	NC03	IN	IT3HX	-	open	157	NC22	IN	IT3HX	-	open
38	M_A[12]	OUT	OT3LLR2X	MA	-	158	M_BEX[4]	OUT	OT3LLR2X	MB	-
39	M_A[13]	OUT	OT3LLR2X	MA	-	159	M_BEX[5]	OUT	OT3LLR2X	MB	-
40	M_A[14]	OUT	OT3LLR2X	MA	-	160	M_BEX[6]	OUT	OT3LLR2X	MB	-
41	M_A[15]	OUT	OT3LLR2X	MA	-	161	M_BEX[7]	OUT	OT3LLR2X	MB	-
44	M_D[20]	IO	B3NNLLR2X	MD	-	164	M_WEX	OUT	OT3LLR2X	-	memr/w
45	M_D[21]	IO	B3NNLLR2X	MD	-	165	M_BSTMX	OUT	OT3LLR2X	-	membst
46	M_D[22]	IO	B3NNLLR2X	MD	-	166	M_XREQ	OUT	OT3LLR2X	-	memreq
47	M_D[23]	IO	B3NNLLR2X	MD	-	167	M_A[03]	OUT	OT3LLR2X	MA	memaddr
48	NC04	IN	IT3HX	-	open	168	NC23	IN	IT3HX	-	open
49	NC05	IN	IT3HX	-	open	169	NC24	IN	IT3HX	-	open
50	M_D[24]	IO	B3NNLLR2X	MD	-	170	M_A[24]	OUT	OT3LLR2X	MA	-
51	M_D[25]	IO	B3NNLLR2X	MD	-	171	M_A[25]	OUT	OT3LLR2X	MA	-
52	M_D[26]	IO	B3NNLLR2X	MD	-	172	M_A[26]	OUT	OT3LLR2X	MA	-
53	M_D[27]	IO	B3NNLLR2X	MD	-	173	M_A[27]	OUT	OT3LLR2X	MA	-
56	M_D[28]	IO	B3NNLLR2X	MD	-	176	M_A[28]	OUT	OT3LLR2X	MA	-
57	M_D[29]	IO	B3NNLLR2X	MD	-	177	M_A[29]	OUT	OT3LLR2X	MA	-
58	M_D[30]	IO	B3NNLLR2X	MD	-	178	M_A[30]	OUT	OT3LLR2X	MA	-
59	M_D[31]	IO	B3NNLLR2X	MD	-	179	M_A[31]	OUT	OT3LLR2X	MA	-
60	NC06	IN	IT3HX	-	open	180	NC25	IN	IT3HX	-	open
61	NC07	IN	IT3HX	-	open	181	NC26	IN	IT3HX	-	open
62	C_D[00]	IO	B3NNLLR2X	CD	regdata	182	NC27	IN	IT3HX	-	open
63	C_D[01]	IO	B3NNLLR2X	CD	-	183	NC28	IN	IT3HX	-	open
64	C_D[02]	IO	B3NNLLR2X	CD	-	184	NC29	IN	IT3HX	-	open
65	C_D[03]	IO	B3NNLLR2X	CD	-	185	NC30	IN	IT3HX	-	open
68	C_D[04]	IO	B3NNLLR2X	CD	-	188	M_D[56]	IO	B3NNLLR2X	MD	-
69	C_D[05]	IO	B3NNLLR2X	CD	-	189	M_D[57]	IO	B3NNLLR2X	MD	-
70	C_D[06]	IO	B3NNLLR2X	CD	-	190	M_D[58]	IO	B3NNLLR2X	MD	-
71	C_D[07]	IO	B3NNLLR2X	CD	-	191	M_D[59]	IO	B3NNLLR2X	MD	-
72	NC08	IN	IT3HX	-	open	192	NC31	IN	IT3HX	-	open
73	C_D[08]	IO	B3NNLLR2X	CD	-	193	M_D[60]	IO	B3NNLLR2X	MD	-
74	C_D[09]	IO	B3NNLLR2X	CD	-	194	M_D[61]	IO	B3NNLLR2X	MD	-
75	C_D[10]	IO	B3NNLLR2X	CD	-	195	M_D[62]	IO	B3NNLLR2X	MD	-
76	C_D[11]	IO	B3NNLLR2X	CD	-	196	M_D[63]	IO	B3NNLLR2X	MD	-
79	C_D[12]	IO	B3NNLLR2X	CD	-	199	M_D[36]	IO	B3NNLLR2X	MD	-
80	C_D[13]	IO	B3NNLLR2X	CD	-	200	M_D[37]	IO	B3NNLLR2X	MD	-
81	C_D[14]	IO	B3NNLLR2X	CD	-	201	M_D[38]	IO	B3NNLLR2X	MD	-
82	C_D[15]	IO	B3NNLLR2X	CD	-	202	M_D[52]	IO	B3NNLLR2X	MD	-
83	NC09	IN	IT3HX	-	open	203	M_D[53]	IO	B3NNLLR2X	MD	-
84	NC10	IN	IT3HX	-	open	204	NC32	IN	IT3HX	-	open
85	C_D[16]	IO	B3NNLLR2X	CD	-	205	M_D[54]	IO	B3NNLLR2X	MD	-
86	C_D[17]	IO	B3NNLLR2X	CD	-	206	M_D[33]	IO	B3NNLLR2X	MD	-
87	C_D[18]	IO	B3NNLLR2X	CD	-	207	M_D[34]	IO	B3NNLLR2X	MD	-
88	C_D[19]	IO	B3NNLLR2X	CD	-	208	M_D[35]	IO	B3NNLLR2X	MD	-
91	C_WEX	OUT	OT3LLR2X	-	regr/w	211	TCK	IN	IT3HX	-	jtag
92	M_D[40]	IO	B3NNLLR2X	MD	-	212	M_D[39]	IO	B3NNLLR2X	MD	-
93	M_D[41]	IO	B3NNLLR2X	MD	-	213	TDI	IN	IT3HX	-	jtag
94	M_D[42]	IO	B3NNLLR2X	MD	-	214	TRST	IN	IT3HX	-	jtag
95	NC11	IN	IT3HX	-	open	215	TMS	IN	IT3HX	-	jtag
96	NC12	IN	IT3HX	-	open	216	TDO	OUT	OT3LLR2X	-	jtag
97	M_D[43]	IO	B3NNLLR2X	MD	-	217	M_D[44]	IO	B3NNLLR2X	MD	-
98	C_XREQ	OUT	OT3LLR2X	-	regreq	218	M_D[45]	IO	B3NNLLR2X	MD	-
99	C_XACK	OUT	OT3LLR2X	-	regack	219	M_D[46]	IO	B3NNLLR2X	MD	-
100	M_D[32]	IO	B3NNLLR2X	MD	-	220	M_D[47]	IO	B3NNLLR2X	MD	-
103	C_D[20]	IO	B3NNLLR2X	CD	-	223	M_D[55]	IO	B3NNLLR2X	MD	-
104	C_D[21]	IO	B3NNLLR2X	CD	-	224	M_A[04]	OUT	OT3LLR2X	MA	-
105	C_D[22]	IO	B3NNLLR2X	CD	-	225	M_D[48]	IO	B3NNLLR2X	MD	-
106	C_D[23]	IO	B3NNLLR2X	CD	-	226	M_D[49]	IO	B3NNLLR2X	MD	-
107	NC13	IN	IT3HX	-	open	227	M_D[50]	IO	B3NNLLR2X	MD	-
108	NC14	IN	IT3HX	-	open	228	M_D[51]	IO	B3NNLLR2X	MD	-
109	NC15	IN	IT3HX	-	open	229	NC33	IN	IT3HX	-	open
110	C_D[24]	IO	B3NNLLR2X	CD	-	230	M_D[00]	IO	B3NNLLR2X	MD	-
111	C_D[25]	IO	B3NNLLR2X	CD	-	231	M_D[01]	IO	B3NNLLR2X	MD	-
112	C_D[26]	IO	B3NNLLR2X	CD	-	232	M_D[02]	IO	B3NNLLR2X	MD	-
113	C_D[27]	IO	B3NNLLR2X	CD	-	233	M_D[03]	IO	B3NNLLR2X	MD	-
116	C_D[28]	IO	B3NNLLR2X	CD	-	236	M_D[04]	IO	B3NNLLR2X	MD	-
117	C_D[29]	IO	B3NNLLR2X	CD	-	237	M_D[05]	IO	B3NNLLR2X	MD	-
118	C_D[30]	IO	B3NNLLR2X	CD	-	238	M_D[06]	IO	B3NNLLR2X	MD	-
119	C_D[31]	IO	B3NNLLR2X	CD	-	239	M_D[07]	IO	B3NNLLR2X	MD	-
120	NC16	IN	IT3HX	-	open	240	NC34	IN	IT3HX	-	open

索引