JUMP-1 2次キャッシュ内部資源仕様書 (ver. 2.0)

JUMP-1 設計グループ

Revision History

1.0	'96 10/30	H. Nakashima	第一次仕様作成
2.0	'97 09/02	H. Nakashima	実装に基づく修正

目次

1　概説

2 次キャッシュの内部資源は，表 1に示すように，Control Space と呼ばれる 32 ビットのアドレス空間中にマップされている。資源のアクセスは，そのビット幅によらず LDAB/STAB（一部は LDB/STB も可）によって行ない，またそのアドレスは doubleword 境界でなければならない。また表の cat は，以下に示す資源のアクセスに関する制限を示したものである。

- (A)
  MCMD = 0001 の LDB/STB によるアクセスを，ユーザに開放することを想定したもの。
- (B)
  MCMD = 0001 の LDB/STB によるアクセスを，ユーザに開放しないことを想定したもの。但し OS の判断によって，開放することもできる。
- (Cd)
  LDAB/STAB など，Control Space をアクセスできる特権命令によってのみアクセスでき，かつ直接ロード／ストアが可能なもの。
- (Ci)
  LDAB/STAB など，Control Space をアクセスできる特権命令によってのみアクセスでき，Cd に属するレジスタ CSADR を経由した間接ロード／ストアのみが可能なもの。表 1のアドレスに対して（LDAB ではなく）STAB を行なうと，資源の値が CSADR に格納される。またアドレスに 0x10000000 を加えたものに対して STAB を行なうと，CSADR の値が資源に格納される。
- (Cx)
  LDAB/STAB など，Control Space をアクセスできる特権命令によってのみアクセスでき，ロード／ストアに特殊な手順が必要なもの。

表 1: 2 次キャッシュ内部資源一覧 (1/5)

cat	ADDR	r/w	width	name	note
A	00000000	w	0	EBAPRV	EB Approve
	00000008			(reserved)	
	00000010	w	0	EBPREQ	EB Pre-Request
	00000018			(reserved)	
	00000020			(reserved)	
	00000028			(reserved)	
	00000030	r	1	EBRREQ	EB Real-Request
	00000038			(reserved)	
	00000040	w	0	WBFLSH	Write Buffer Flush
	00000048			(reserved)	
	00000050			(reserved)	
	00000058			(reserved)	
	00000060	w	0	L2DMY	Dummy Register
	00000068			(reserved)	
	⋮				
	00000FF8			(reserved)	
B	00001000	r/w	8	EBCAP	EB Approve Counter
	00001008			(reserved)	
	00001010	r/w	8	EBCPR	EB Pre-Request Counter
	00001018			(reserved)	
	00001020	r/w	8	EBCCP	EB Completed Request Counter
	00001028			(reserved)	
	00001030	w	0	EBCLR	EB Clear
	00001038			(reserved)	
	00001040	r/w	2	INJLEV	Injection Level
	00001048			(reserved)	
	00001050	r	2	PEID	Processor Element ID
	00001058			(reserved)	
	00001060			(reserved)	
	⋮				
	00001FF8			(reserved)	
	00002000			(reserved)	
	⋮				
	0FFFFFF8			(reserved)	

表 1: 2 次キャッシュ内部資源一覧 (2/5)

cat	ADDR	r/w	width	name	note
Cd	10000000	r/w	8	CSADR[31:24]	Cont Space Access Data
	10000008	r/w	8	CSADR[23:16]	
	10000010	r/w	8	CSADR[15:8]	
	10000018	r/w	8	CSADR[7:0]	
	10000020			(reserved)	
	⋮				
	10000078			(reserved)	
	10000080	r	4	BMPRDR[11:8]	Backmap Read Data
	10000088	r	8	BMPRDR[7:0]	
	10000090	r/w	4	L2CCTL[11:8]	L2C Control Reg
	10000098	r/w	8	L2CCTL[7:0]	
	100000A0	r/w	2	L2CFRZ	L2C Freeze
	100000A8			(reserved)	
	100000B0	r	1	L2CERR	L2C Error
	100000B8			(reserved)	
	100000C0	r/w	6	ERRST[13:8]	Access Error Status
	100000C8	r/w	8	ERRST[7:0]	
	100000D0	r	7	FMPEAR[14:8]	Frotmap Error Address
	100000D8	r	8	FMPEAR[7:0]	
	100000E0			(reserved)	
	100000E8			(reserved)	
	100000F0	r	8	INTST	Bus Interrupt Status
	100000F8			(reserved)	
	10000100	w	8	INTRST	Reset Interrupt Status
	10000108			(reserved)	
	10000110	w	8	INTSET	Set Interrupt Status
	10000118			(reserved)	
	10000120	w	2	SNDINV	SNDBUF invalidate
	10000128			(reserved)	
	10000130	w	2	RCVINV	RCVBUF invalidate
	10000138			(reserved)	
	10000140	w	1	EMMTRP	Emergency Trap to pseudo MBP
	10000148			(reserved)	
	10000150	w	1	L2CRDY	Make CPU accessible to L2C
	10000158			(reserved)	
	10000160	w	4	ECSAID	External Cont Space Access ID
	10000168			(reserved)	
	10000170			(reserved)	
	⋮				
	3FFFFFF8			(reserved)	

表 1: 2 次キャッシュ内部資源一覧 (3/5)

cat	ADDR	r/w	width	name	note
Cd	40000000	r/w	64	DARRAY[0]	L2C Data Array
	⋮				
	400FFFF8	r/w	64	DARRAY[0xFFFF8]	
	40100000			(reserved)	
	⋮				
	4FFFFFF8			(reserved)	
	50000000			(reserved)	
	⋮				
	5FFFFFF8			(reserved)	
	60000000	w	0	INV[0]	Invalidate Cache Entry
	60000020	w	0	INV[1]	
	⋮				
	600FFFE0	w	0	INV[0x7FFF]	
	60100000			(reserved)	
	⋮				
	6FFFFFF8			(reserved)	
	70000000			(reserved)	
	⋮				
	7FFFFFF8			(reserved)	

表 1: 2 次キャッシュ内部資源一覧 (4/5)

cat	ADDR	r/w	width	name	note
Ci	80000000			(reserved)	
	⋮				
	80000038			(reserved)	
	80000040	r/w	32	RCVDIR[0][63:32]	RCVBUF directory
	⋮				
	80000058	r/w	32	RCVDIR[3][63:32]	
	80000060	r/w	32	RCVDIR[0][31:0]	
	⋮				
	80000078	r/w	32	RCVDIR[3][31:0]	
	80000080	r/w	32	SNDDIRA[0][63:32]	SNDBUF directory (part-A)
	⋮				
	80000098	r/w	32	SNDDIRA[3][63:32]	
	800000A0	r/w	32	SNDDIRA[0][31:0]	
	⋮				
	800000B8	r/w	32	SNDDIRA[3][31:0]	
	800000C0	r/w	8	SNDDIRB[0][39:32]	SNDBUF directory (part-B)
	⋮				
	800000D8	r/w	8	SNDDIRB[3][39:32]	
	800000E0	r/w	32	SNDDIRB[0][31:0]	
	⋮				
	800000F8	r/w	32	SNDDIRB[3][31:0]	
	80000100	r/w	14	REQRCVQ	Request Receive Queue
	80000108	r/w	14	REPRCVQ	Reply Receive Queue
	80000110	r/w	14	REQSNDQ	Request Send Queue
	80000118	r/w	14	REPSNDQ	Reply Send Queue
	80000120			(reserved)	
	80000128	r/w	32	RETBUF	Retry Buffer
	80000130	r/w	4	SDWORK[35:32]	SNDDIR Work Register
	80000138	r/w	32	SDWORK[31:0]	
	80000140	r/w	32	TOCTR	Timeout Counter
	80000148	r/w	32	TOLMT	Timeout Limit
	80000150	r/w	16	WMCTR	Write Merge Timeout Counter
	80000158	r/w	16	WMLMT	Write Merge Timeout Limit
	80000160	r/w	24	PFCTR	Prefetch Counters
	80000168	r/w	12	PFLMT	Prefetch Limit
	80000170			(reserved)	
	80000178			(reserved)	
	80000180	r/w	19	MESPATB	Measurement Pattern for Bus Event
	80000188	r/w	19	MESMSKB	Mask for MESPATB
	80000190	r/w	32	MESCTRB	Counter for MESPATB
	80000198			(reserved)	
	800001A0	r/w	16	MESPATP	Measurement Pattern for Proc Event
	800001A8	r/w	16	MESMSKP	Mask for MESPATP
	800001B0	r/w	32	MESCTRP	Counter for MESPATP
	800001B8			(reserved)	
	800001C0			(reserved)	
	⋮				
	8FFFFFF8			(reserved)	

表 1: 2 次キャッシュ内部資源一覧 (5/5)

cat	ADDR	r/w	width	name	note
Ci	90000000				for store 80000000–8FFFFFF8
	⋮				
	9FFFFFF8				
	A0000000	r/w	28	FMP[0]	L2C Cache Tag Front Map
	A0000020	r/w	28	FMP[1]	
	⋮				
	A00FFFE0	r/w	28	FMP[0x7FFF]	
	A0100000			(reserved)	
	⋮				
	AFFFFFF8			(reserved)	
	B0000000				for store A0000000–AFFFFFF8
	⋮				
	BFFFFFF8				
	C0000000			(reserved)	
	⋮				
	DFFFFFF8			(reserved)	
Cx	E0000000	w	0	RBMP[0]	Read Cache Tag Back Map
	E0000020	w	0	RBMP[1]	
	⋮				
	E01FFFE0	w	0	RBMP[0x7FFF]	
	E0200000			(reserved)	
	⋮				
	EFFFFFF8			(reserved)	
	F0000000	w	0	WBMP[0](0)	Write 0 to L2C Cache Tag Back Map
	F0000020	w	0	WBMP[1](0)	
	⋮				
	F00FFFE0	w	0	WBMP[0x7FFF](0)	
	F0100000	w	0	WBMP[0](1)	Write 1 to L2C Cache Tag Back Map
	F0100020	w	0	WBMP[1](1)	
	⋮				
	F01FFFE0	w	0	WBMP[0x7FFF](1)	
	F0200000	w	0	WBMP[0](1)	
	⋮				
	FFEFFFE0	w	0	WBMP[0x7FFF](0xFE)	
	FFF00000	w	0	WBMP[0](0xFF)	Write 0xFF to L2C Cache Tag Back Map
	FFF00020	w	0	WBMP[1](0xFF)	
	⋮				
	FFFFFFE0	w	0	WBMP[0x7FFF](0xFF)	

2 内部資源の仕様

EBAPRV (`0x00000000`)

任意の値を書くと Elastic Barrier の同期点到達を報告する。なお将来，その同期が成立したことを確認してはならない。

EBPREQ (`0x00000010`)

任意の値を書くと Elastic Barrier の同期点到達を報告する。なお将来，その同期が成立したことを確認しなければない。

EBRREQ (`0x00000030`)

Elastic Barrier の未確認同期点の中で，最も古いものが成立していれば-1, 成立していなければ 0 を返す。

[初期値] 不定である。EBCCP を 0 に初期化することにより，間接的に 0 に初期化される。

WBFLSH (`0x00000040`)

任意の値を書くと，書込データのマージのためのバッファに残っている全ての書込要求について，バスへの送出を開始する。書込要求のバスへの送出が完了したことは，STBAR で間接的に確認することができる。

L2DMY (`0x00000060`)

任意の値を書いても，システムの状態を一切変更しないダミー・レジスタ。prefetch/cflush や **bq-write** に関する制限を確実に守るために使用することができる。

EBCAP (`0x00001000`)

Elastic Barrier の未成立同期点の内，EBAPRV で報告したものの数を保持するカウンタ。

[初期値] 不定であるので 0 に初期化する。

EBCPR (`0x00001010`)

Elastic Barrier の未成立同期点の内，EBPREQ で報告したものの数を保持するカウンタ。

[初期値] 不定であるので 0 に初期化する。

EBCCP (`0x00001020`)

Elastic Barrier の成立した同期点の内，EBRREQ で確認していないもの数を保持するカウンタ。

[初期値] 不定であるので 0 に初期化する。

EBCLR (`0x00001030`)

任意の値を書くと，Elastic Barrier に関するカウンタ EBCAP, EBCPR, EBCCP，並びに関連する上位階層が持つ全てのカウンタをクリアする。クリア動作の完了は，EBRREQ で確認しなければならない。

INJLEV (`0x00001040`)

Injection を実行する条件を，以下のように定める。

- INJLEV $= 0$
  該当するラインが無効であり，かつ直前に有効であった時には同じアドレスのデータが存在した場合に実行。
- INJLEV $= 1$
  該当するラインが無効であれば実行。

- INJLEV = 2
  該当するラインが無効であるか，既に存在するデータをクラスタ・メモリへ書き戻す必要がない場合に実行。
- INJLEV = 3
  無条件に実行。

[初期値] 0 である。必要に応じて適切な値に初期化する。

PEID (`0x00001050`)
クラスタ内のどの PE であるかを，$0 \sim 3$ の値で返す。

[初期値] 適切な値である。初期化不要。

CSADR (`0x00001000`... `38`)
表 1の cat が Ci であるような 2 次キャッシュ資源の，読出 / 書込に用いる。

[初期値] 不定であるが初期化不要。

BMPRDR (`0x000001080`... `88`)
バックマップ読出 (RBMP) を行なった結果として読出された値を保持する。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				P	V	G	O	ADDR							

- P = BMPRDR[11] (Parity)
  BMPRDR[10:0] に対する偶数パリティ。
- V = BMPRDR[10] (Valid)
  1 であれば，バックマップ・エントリが有効であることを示す。
- G = BMPRDR[9] (Global)
  1 であれば，バックマップ・エントリに保持されたブロックが，クラスタ間で共有されている（可能性がある）ことを示す。
- O = BMPRDR[8] (Owned)
  1 であれば，バックマップ・エントリに保持されたブロックに対する，他のキャッシュまたは MMC/MBPL からの読出要求に応答する責任があることを示す。
- BA = BMPRDR[7 : 0] (Block Address)
  バックマップ・エントリに保持されたブロックのアドレスを a とした時，$a[27:20]$ を示す。

[初期値] 不定であるが初期化不要。

L2CCTL (`0x10000090`... `98`)
2 次キャッシュの動作モードを，以下のように定める。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				DLYPF	0	WMQDP		FMPPEN	BMPPEN	WACKS	WACKQ	WACKBQ	L2CCBL	CTYPE	

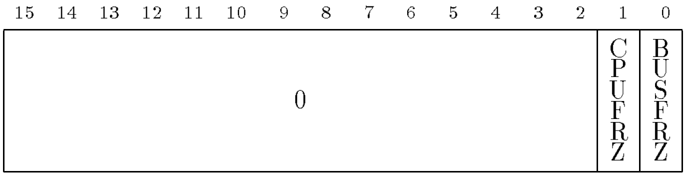
- DLYPF = L2CCTL[11] (Delayed Prefetch)
  1 であれば，プリフェッチによって獲得したラインの 2 次キャッシュ・データアレイへの書込を，CPU がそのラインをアクセスするまで遅らせる。

- WMQDP = L2CCTL[9 : 8] (Write Merge Queue Depth)
  書込データ・マージのためのキューの深さ ($1 \leq \mathrm{WMQDP} \leq 3$)。
- FMPPEN = L2CCTL[7] (Frontmap Parity Enable)
  1 であれば，キャッシュ・タグのフロント・マップのパリティ・チェックを行なう。
- BMPPEN = L2CCTL[6] (Backmap Parity Enable)
  1 であれば，キャッシュ・タグのバック・マップのパリティ・チェックを行なう。
- WACKS = L2CCTL[5] (With Acknowledge for s-write)
  1 であれば，**s-write** 操作完了を STBAR で確認することができる。通常は **s-write** の完了を確認する必要はなく，性能上は 0 であることが望ましい。
- WACKQ = L2CCTL[4] (With Acknowledge for q-write)
  1 であれば，**q-write** 操作完了を STBAR で確認することができる。通常は **q-write** の完了を確認する必要はなく，性能上は 0 であることが望ましい。
- WACKBQ = L2CCTL[3] (With Acknowledge for bq-write)
  1 であれば，**bq-write** 操作完了を STBAR で確認することができる。通常は **bq-write** の完了を確認する必要はなく，性能上は 0 であることが望ましい。
- L2CCBL = L2CCTL[2] (L2C Cachable for Special Commands)
  1 であれば，1 次キャッシュ不可とされた一部のメモリ・コマンドを，2 次キャッシュ可で実行する。
- CTYPE = L2CCTL[1 : 0] (Command Type)
  4 種類 ($0 \sim 3$) のメモリ・コマンド・セットのいずれを用いるかを定める。

[初期値] WMQDP = 1, CTYPE = 11, その他はである。必要に応じて適切な値に初期化する。

L2CFRZ (`0x100000A0`)

例外情報獲得などのために，2 次キャッシュをフリーズする。



- CPUFRZ = L2CFRZ[1] (CPU access freeze)
  1 であれば CPU からのアクセスを拒否し，アクセスがあった場合には CPUFRZ が 0 になるまで待たせる。従って，CPUFRZ を CPU が 1 にすることは，極めて危険である。
- BUSFRZ = L2CFRZ[0] (BUS access freeze)
  1 であれば共有バスに関する送受信を行なわず，Control Space を除く全ての CPU からのアクセスを拒否する（アクセスがあると bus-error となる）。

[初期値] 0 である。初期化不要。

L2CERR (`0x100000B0`)

2 次キャッシュ内部で例外を検出すると 1 になる。L2CERR が 1 である間は，完了していない書込操作が存在していても，全て完了したかのように見える。

[初期値] 0 である。初期化不要。

ERRST (`0x100000C0` . . . `C8`)

2 次キャッシュが検出した例外に関する詳細な情報を保持する。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEP	MBE	MUC	MTO	EXT	EP	BECMD	BECBL	BESPC	BEFRZ	UCFMP	UCBMP	TO	0	TOENT	

- MEP = ERRST[15] (Multiple Packet Error)
  ERRST が不正パケット受信が生起したことを示している間に（即ち EP=1），再び不正パケットを受信したことを示す。
- MBE = ERRST[14] (Multiple Bus Error)
  ERRST がある原因による bus-error が生起したことを示している間に（即ち BExxx=1），同じ原因による bus-error が生起したことを示す。このような状況は通常，2 次キャッシュが CPU に例外を通知して CPU が trap routine に移行するまでの間に行なわれる store buffer の掃き出しが，新たな例外を起こしたことを意味する。なお別の原因による bus-error が重複して生起した場合には，MBE は 1 にはならず，単に二つの BExxx が 1 になるだけである。
- MUC = ERRST[13] (Multiple Uncorrectable Error)
  ERRST が uncorrectable-error が生起したことを示している間に（即ち UCxxx=1），同じ原因による uncorrectable-error が生起したことを示す。この状況では最後の uncorrectable-error に関する情報のみが保存されている。なお別の原因による Uncorrectable-error が重複して生起した場合には，MUC は 1 にはならず，単に二つの UCxxx が 1 になるだけである。
- MTO = ERRST[12] (Multiple Timeout Error)
  ERRST が time-out が生起したことを示している間に（即ち TO=1），再び time-out が生起したことを示す。この状況では最後の time-out に関する情報のみが保存されている。
- EXT = ERRST[11] (External Asynchronous Error)
  INTST[7] を 1 にするような割込要求を，CPU, クラスタ内の他のプロセッサ，または MMC/MBPL が行なったことを示す。EXT は他の例外ビットと同時に 1 になることがある。
- EP = ERRST[10] (Error Packet)
  不正なパケットあるいは *error_reply* を受信したことを示す。*error_reply* に対応する CPU のアクセスが read または swap であれば，エラーの通知は uncorrectable-error によって同期的に行なわれ，それ以外の場合は NMI によって非同期的に行なわれる。

以下の 4 ビットは bus-error の原因を示す。なお，特に非同期的と記されたものは NMI により通知される。通常は排他的であるが，MBE の項で述べた原因によって二つのビットが 1 になることがある。また他の例外ビットと同時に 1 になることもある。

- BECMD = ERRST[9] (Bus Error on Illegal Command)
  定義されていないメモリ・コマンドを実行しようとしたことを示す。
- BECBL = ERRST[8] (Bus Error on Illegal Cachability)
  キャッシュ可否に関する制限に反して，メモリ・コマンドを実行しようとしたことを示す。
- BESPC = ERRST[7] (Bus Error on Command Specific Exception)
  以下に示すような，個々のメモリ・コマンドに特有の例外が発生したことを示す。
  - **prefetch/cflush** を doubleword 境界以外のアドレスに対して行なった。
  - **rg-read/rg-write** を doubleword 境界以外のアドレスに対して行なった。
  - **prefetch/cflush** に関する同一ブロックへの連続書込禁止規則が破られ，かつそれが致命的である。通知は非同期的である。
  - アドレス変換の過程で **s-read/f-read** を実行しようとした。通知は非同期的である。

  - **q-read/p-read/q-check** の実行時に，キャッシュ・ブロックが他のプロセッサ／クラスタに共有されていることが判明した。通知は非同期的である。
  - **bq-read/bq-check/bq-write** の実行時に，キャッシュ・ブロックが他のプロセッサ／クラスタに共有されていることが判明した。通知は非同期的である。
  - **bq-wirte** に関する同一ブロックへの連続書込禁止規則が破られ，かつそれが致命的である。通知は非同期的である。
  - **bq-read/bq-check** の実行過程で，不正な読出権移動が生じた。

- BEFRZ = ERRST[6] (Bus Error on Freezed L2C Access)
  L2CFRZ.BUSFRZ が 1 である間に，CPU が通常のメモリ空間をアクセスしようとしたことを示す。この例外はほとんどの場合致命的であり，CPU の watchdog reset を引き起こす可能性が高い。

以下の 2 ビットは，uncorrectable-error の原因を示す。

- UCFMP = ERRST[5] (Uncorrectable Error on Frontmap Bad Parity)
  フロントマップのパリティエラーが原因で，uncorrectable-error が生起したことを示す。なお，フロントマップ・パリティエラーは同期的（即ち命令フェッチや load 系命令の結果として）にも非同期的にも発生し，いずれの場合も UCFMP が 1 となる。CPU が受理したトラップが NMI であれば非同期的であり，そうでない場合は同期的である。またエラーを起こしたアドレスは FMPEAR に保持される。
- UCBMP = ERRST[4] (Uncorrectable Error on Backmap Bad Parity)
  バックマップのパリティエラーが原因で，uncorrectable-error が生起したことを示す。なお，バックマップ・パリティエラーは必ず非同期的に発生する。

以下のビットは，time-out の原因を示す。

- TO = ERRST[3] (Timeout Error)
  time-out が生起したことを示す。なお，time-out は同期的（即ち命令フェッチや load 系命令の結果として）にも非同期的にも発生し，いずれの場合も TO が 1 となる。CPU が受理したトラップが interrupt-level-15 であれば非同期的であるが，そうでない場合に同期的であるとは限らず，CPU 内部に保存された例外原因アドレス（論理アドレス）と，後述する SNDDIR に保存されたアドレス（物理アドレス）とを比較して判断する必要がある。
- TOENT = ERRST[1 : 0] (Timeout Entry of SNDDIR)
  time-out を起こした物理アドレスを含む情報は，8 エントリの SNDDIR のいずれかに保存されている。TOENT は情報が保存されている SNDDIR のエントリ番号を示す。

[初期値] 0 である。初期化不要。

FMPEAR (`0x100000D0`...`D8`)
最後にフロントマップ・パリティ・エラーを検出したアドレスを保持する。

[初期値] 不定であるが初期化不要。

INTST, INTRST, INTSET (`0x100000F0,100,110`)
INTST は，クラスタ内の他のプロセッサまたは MMC/MBPL が発行した割込要求を保持するレジスタであり，INTST[7] (MSB) がレベル 15 の割込（NMI, マスク不能）に，INTST[6] がレベル 14 の割込（マスク可能）に，以下順に INTST[i] ($i \geq 0$) がレベル $i+8$ の割込（マスク可能）に対応する。なお，INTST[7] は 2 次キャッシュが検出した非同期的なアクセス例外によっても 1 になる。またレベル 7 以下の割込が報告されることはない。

INTST[7:0] のいずれかが 1 であると，1 であるものの中でもっともレベルが高い割込がプロセッサに通知される。また割込の通知は対応する INTST のビットを 0 にするまで，継続的に行なわれる。なお既に 1 であるビットに対して重複して割込要求がなされても，重複割込の通知などの特別な操作は一切行なわれない。

INTST の更新は，INTRST または INTSET へのストアによって行なわれる。INRST へのストアは，通常の意味のストアではなく，ストア・データの論理否定と INTST の値との論理積を INTST に書込むことを意味する。つまり INTST[i] を 0 にする（レベル $i+8$ の割込要因消去）は，2^iを INTST にストアすることによって実現される。同様に INTSET へのストアは，ストア・データと INTST の値との論理和を INTST に書込むことを意味する。他の PE の INTSET へのストアにより，割込要求を行なうことができる。

[初期値] INTST $= 0$ である。初期化不要。

SNDINV (`0x10000120`)
n をストアすると SNDDIR[n] を無効化する。

RCVINV (`0x10000130`)
n をストアすると RCVDIR[n] を無効化する。

EMMTRP (`0x10000140`)
1 を書くと，MMC/MBPL に対して緊急トラップを発行する。このトラップはメモリ機構をバイパスして伝えられる。0 を書くとトラップが解除される。

[初期値] 0 である。初期化不要。

L2CRDY (`0x10000150`)
1 を書込むと CPU によるメモリ・システムのアクセスが可能になる。リセット時には 0 であり，CPU のリセット割込ルーチンの実行を抑止している。2 次キャッシュを含むメモリ・システムの初期化後，MBPL などから L2CRDY を 1 にすることにより，リセット割込ルーチンの実行を開始することができる。0 の書込は極めて危険な操作で，実行される状況は想定されておらず，また実行の結果は全く保証されない。

[初期値] 0 である。MMC/MBPL より 1 に初期化する。

ECSAID (`0x10000160`)
ECSAID に下表に示す値をストアすると，その次に行なわれる Control Space へのストアが，他のプロセッサあるいは MMC/MBPL へのアクセスとして解釈される。

ECSAID	function
0x0	(reserved)
⋮	⋮
0x7	(reserved)
0x8	store PE0 resource
0x9	store PE1 resource
0xA	store PE2 resource
0xB	store PE3 resource
0xC	store MMC/MBPLresource
0xD	(reserved)
0xE	(reserved)
0xF	store PE0-3, MMC/MBPLresources

- store PEn resource
  次に行なわれるストアにより，データが同一クラスタに属する他のプロセッサ (PEn) の 2 次キャッシュ資源に書込まれる。資源のアドレスは次のストアのアドレスである。
- store MMC/MBPLresource
  次に行なわれるストアにより，データが同一クラスタに属する MMC/MBPL の資源に書込まれる。ストアのアドレスと資源との対応関係は，別途定められる。
- store PE0-3, MMC/MBPLresources
  次に行なわれるストアにより，データが同一クラスタに属する全てのプロセッサの 2 次キャッシュ資源と，MMC/MBPL の資源に同時に書込まれる。

[初期値] 0 であり，かつ ECSAID へのストアが行なわれていない状態にある。初期化不要。

DARRAY[n] (`0x40000000` . . . `400FFFF8`)

2 次キャッシュのデータ・アレイ。

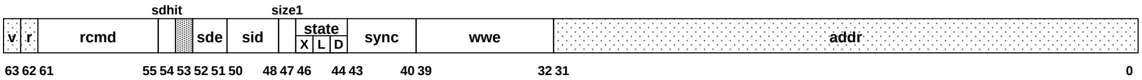
[初期値] 不定であるが，FMP を初期化すればよい。

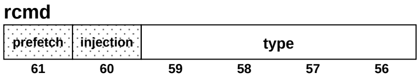
INV[n] (`0x60000000` . . . `600FFFE0`)

任意の値を書くと，2 次キャッシュのブロック $n = a[19:5]$ を無条件に無効化し，必要であればクラスタ・メモリへの書き戻しを行う。この操作の対象ブロックに対して他のプロセッサがアクセスを並行して行なうと，不正な結果が生じることがある。

RCVDIR[n] (`0x80000040` . . . `78`)

共通バスから入力したパケットを処理するための情報を保持する。





- v = RCVDIR[63] (Valid)
  1 であれば，エントリが有効であることを示す。
- r = RCVDIR[62] (Reserved Entry)
  1 であれば，エントリが予約済であることを示す。
- rcmd.prefetch = RCVDIR[61] (Prefetch)
  1 であれば，プリフェッチ要求（block queue のプリフェッチを含む）への応答である。
- rcmd.injection = RCVDIR[60] (Injection)
  1 であれば，インジェクション要求（allread_reply によるものを含む）または *control_write_req*。
- rcmd.type[3:0] = RCVDIR[59 : 56]
  要求／応答の種類を示す。

  $0 =$ *read_req/read_reply*
  $1 =$ *rdinv_req/rdinv_reply*
  $2 =$ *allread_req/allread_reply*
  $3 =$ *swap_req/*reply for prefetch
  $4 =$ *update_xxx_req/bq_nop_reply*
  $5 =$ *noalloc_write_req*
  $7 =$ *bq_nop_reply* for pretch

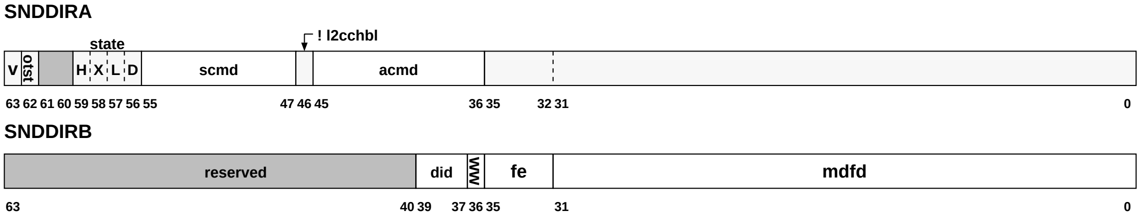
$8 =$ 大域化のための疑似要求$/control_write_req$
$9 =$ 非共有状態への遷移のための疑似要求
$10 = invalidate\_req/bq\_nop\_reply$
$11 = cflush\_req$
$12 = demap\_req$
$13 = control\_read\_req$
$14 =$ pseudo-reply for swap
$15 = swap\_req$ (for non-owner)$/error\_reply$

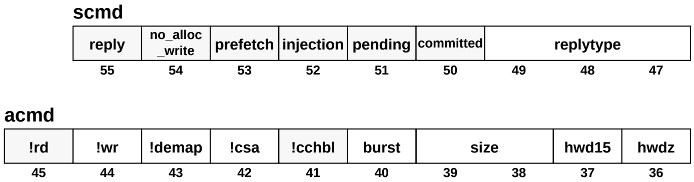
- sdhit = RCVDIR[55] (SNDDIR hit)
  1 であれば，応答／要求と同じアドレスに関する要求が SNDDIR に存在する。
- sde[1:0] = RCVDIR[53 : 52] (SNDDIR Entry Number)
  応答／要求に対応する要求を保持している SNDDIR のエントリ番号。
- sid[2:0] = RCVDIR[50 : 48] (Source ID)
  パケットの送信元。
- size1 = RCVDIR[47] (Size)
  パケットのアクセス幅。可能な区別は以下の 2 種類である。
  - 0 : block,　1 : doubleword
  - 0 : byte,　1 : word
- state.X = RCVDIR[46] (Exclusive)
  1 であれば，応答されたラインを他の PE は共有していない。
- state.L = RCVDIR[45] (Local)
  1 であれば，応答されたラインを他のクラスタは共有していない。
- state.D = RCVDIR[44] (Dirty)
  1 であれば，応答されたラインの値がクラスタ・メモリに反映されていない。
- fe[3:0] = RCVDIR[43 : 40] (Full/Empty)
  fe[i] は，addr[4:3] が i であるような doubleword の full/empty ビット。
- wwe[7:0] = RCVDIR[39 : 32] (Word Write Enable)
  wwe[i] は，アドレスのビット [4:2] が $2 \times \mathrm{addr}[4:3] + i \pmod 8$ であるような word の書込イネーブル。または wwe[$2i + 1$] は，アドレスのビット [4:3] が $\mathrm{addr}[4:3] + i \pmod 4$ であるような doubleword の Write Waiting ビット。
- addr[31:0] = RCVDIR[31 : 0] (Addresss)
  パケットが対象とするアドレス。

[初期値] v = 0, その他は不定である。初期化不要。

SNDDIRA[n], SNDDIRB[n] (`0x80000080` ... `F8`)

共通バスに出力するパケットを生成するための情報を保持する。





- v = SNDDIRA[63] (Valid)
  1であれば，エントリが有効であることを示す。
- otst = SNDDIRA[62] (Output State)
  1であれば，送信済 (応答待ち) の要求であることを示す。
- state.H = SNDDIRA[59] (Tag Hit)
  1であれば，キャッシュ・タグにヒットしたことを示す。
- state.X/L/D = SNDDIRA[58 : 56] (Tag state)
  キャッシュ・タグのX (Exclusive), L (Local), D (Dirty) ビット。
- scmd.reply = SNDDIRA[55] (Reply)
  1であれば応答，0であれば要求。
- scmd.no_alloc_write = SNDDIRA[54] (No-Alloc Write)
  1であれば*no_alloc_write_req*。
- scmd.prefetch = SNDDIRA[53] (Prefetch)
  1であればプリフェッチ要求 (doubleword/block queueのプリフェッチを含む)。
- scmd.injection = SNDDIRA[52] (Injection)
  1であればインジェクション用の疑似応答。
- scmd.pending = SNDDIRA[51] (Pending Write)
  1であれば完了確認が必要な要求。
- scmd.commited = SNDDIRA[50] (Write Merge Committed)
  1であれば書込マージの対象外。
- scmd.replytype = SNDDIRA[49 : 47] (Reply Type)
  応答系の場合は；

  0 = read_reply
  1 = rdinv_reply
  2 = allread_reply
  3 = swap_reply
  4 = control_read_reply
  5 = writeback_reply
  6 = バックマップの無効化
  7 = バックマップの無効化を伴うwriteback_reply

要求系の場合は；

0 = MCMD などによる要求生成（4 を除く）
1 = *control_read_req*
4 = block 単位の書込／プリフェッチ要求

- !l2cchbl = SNDDIRA[46]
  0 であれば 2 次キャッシュ可。
- acmd.!rd = SNDDIRA[45] (Read)
  0 であれば load または swap 系による要求。
- acmd.!wr = SNDDIRA[44] (Write)
  0 であれば store または swap 系による要求。
- acmd.!demap = SNDDIRA[43] (Demap)
  0 であれば demap の完了応答。
- acmd.!csa = SNDDIRA[42] (Control Space Access)
  0 であればバックマップまたは他 PE/MMC/MBPL の control space のアクセス要求。
- acmd.!cchbl = SNDDIRA[41] (Cachable)
  0 であれば 1 次キャッシュ可。
- acmd.burst = SNDDIRA[40] (Busrt Access)
  1 であれば CPU からのバースト・アクセス（ブロック・アクセス）による要求。
- acmd.size = SNDDIRA[39 : 38] (Access Size)
  CPU がアクセスしたバイト長であり；

  0 = byte　1 = halfword　2 = word　3 = doubleword
- acmd.hwd15 = SNDDIRA[37] (Halfword Data [15])
  halfword 書込データのビット 15。
- acmd.hwdz = SNDDIRA[36] (Halfword Data [14:0] is not Zero)
  0 であれば halfword 書込データの [14:0] が 0。
- addr[35:0] = SNDDIRA[35 : 0] (Address)
  物理アドレス。
- did = SNDDIRB[39 : 37] (Destination ID)
  応答の宛先。
- ww = SNDDIRB[36] (Write Waiting)
  キャッシュ・タグの WW ビット。
- fe[3:0] = SNDDIRB[35 : 32] (Full/Empty)
  キャッシュ・タグの FE ビット。
- mdfd[31:0] = SNDDIRB[31 : 0] (Modified)
  $\mathrm{mdfd}[i]$ が 1 であれば，$\mathrm{addr}[35:5] \times 32 + i$ のバイトが更新されている。

[初期値] valid = 0, その他は不定である。初期化不要。

REQRCVQ, REPRCVQ, REQSNDQ, REPSNDQ (`0x80000100 ... 18`)

要求受信キュー，応答受信キュー，要求送信キュー，応答送信キューと，それぞれの制御情報である。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CTR			RPTR		WPTR		E[3]		E[2]		E[1]		E[0]	

- CTR = [14 : 12] (Queue Counter)
  キューに登録されている有効なエントリ数 n $(0 \leq n \leq 4)$ を示す。
- RPTR = [11 : 10] (Queue Read Pointer)
  次にキューから取り出されるエントリの番号を示す。但し REQSNDQ では，タイムアウト検出の参照ポインタである。
- WPTR = [9 : 8] (Queue Write Pointer)
  次にキューに入れられる情報が格納されるエントリの番号を示す。但し REQSNDQ では，書込データのマージ対象の最初のエントリの番号を示す。
- E[i] = [$2i+1 : 2i$] (Buffer Entry)
  i 番目のエントリ。値は SNDBUF/RCVBUF のエントリ番号である。

[初期値] E[i] は不定，その他は 0 である。初期化不要。

RETBUF (0x80000128)

リトライされる可能性がある送信要求を保持する。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A[3]	C[3]			?	E[3]			A[2]	C[2]			?	E[2]		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A[1]	0	C[1]		?	0	E[1]		A[0]	0	C[0]		?	0	E[0]	

- A[i] = RETBUF[$8i+7$] (Active)
  1 であればエントリ i が有効であることを示す。
- C[i] = RETBUF[$8i+6 : 8i+4$] (Counter)
  エントリ i の要求に関する，リトライの有無が確定するまでに要するクロック数を示す。
- E[i] = RETBUF[$8i+1 : 8i+0$] (SNDBUF Entry)
  エントリ i の要求に対応する SNDBUF のエントリ番号。

[初期値] A[i] = 0, C[i] = 111, その他は不定である。初期化不要。

SDWORK (0x80000130... 38)

書込データのマージに用いるスクラッチ・パッド。

- fe[3:0] = SDWORK[35 : 32] (Full/Empty)
  キャッシュ・タグの FE ビット。
- mdfd[31:0] = SDWORK[31 : 0] (Modified)
  mdfd[i] が 1 であれば，addr[31 : 5] + i のバイトが更新されている。

[初期値] 不定であるが初期化不要。

TOCTR (0x80000140)

メモリ・コマンドのタイムアウトを監視するタイマー。

[初期値] 0xFFFFFFFF である。初期化不要。

TOLMT (0x80000148)

メモリ・コマンドがタイムアウトする時間を定める。タイムアウトの検出は，TOLMT の値を L としたとき

- CPU が発行してからの時間が L に満たないメモリ・コマンドは，タイムアウトとはならない。
- CPU が発行してからの時間が $4 \times L$ を超えるメモリ・コマンドは，必ずタイムアウトとなる。

という規則により行なわれる。TOLMT は 32 ビット幅であるので，最大値は約 86 秒である。

[初期値] 0xFFFFFFFF である。初期化不要。

WMCTR, WMLMT (`0x80000150...58`)

共有バスへの書込要求が，書込データのマージのためのバッファに滞在する時間を計測するタイマーと，滞在時間の限界値。正確には，バッファへの投入間隔が WMLMT を越えると，バッファ中の全ての書込要求のバスへの送出を開始する。WMLMT は 16 ビット幅であるので，最大値は約 1.3 ms である。

[初期値] WMLMT = 0xFFFF, WMCTR は不定である。WMCTR を 0 に初期化するのが望ましい。

PFCTR (`0x80000160`)

プリフェッチ要求に対する応答が，バッファに滞在する時間を計測するタイマー。PFCTR$[4i+3:4i]$ と PFCTR[23:16] を結合した（PFCTR[23:16] が下位）12 ビットのタイマーが，RCVDIR$[i]$ に格納されたプリフェッチ応答の滞在時間を計測する。

[初期値] PFCTR[15 : 0] = 0xFFFF, PFCTR[23:16] は不定である。PFCTR[23:16] を 0 に初期化するのが望ましい。

PFLMT (`0x80000168`)

プリフェッチ要求に対する応答が，バッファに滞在する時間の限界値。PFLMT[11:8] は 0xF (15) であってはならないので，最大値は約 77 μs である。

[初期値] PFLMT[11 : 8] = 0, PFLMT[7:0] は不定である。適切な値に初期化する。

MESPATB, MESMSKB, MESCTRB (`0x80000180...90`)

共有バスに流れたパケットの種別，対応するキャッシュ・ラインのステータス，パケットの受理状況が MESPATB に等しければ，MESCTRB がインクリメントされる。なお比較の際に，MESMSKB によるマスクが施される。即ち MESPATB のビット i との比較結果を $e[i]$ とすると，$\prod(e[i]+\overline{\mathrm{MESMSKB}[i]})$ が真であれば MESCTRB がインクリメントされる。MESCTRB は 32 ビット幅であるので，約 86 秒分のバス事象の集計を行なうことができる。MESPATB の各ビットの意味は以下の通り。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0													SCMD		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCMD				ACC	DID			RET	SID			BMPH	BMPV	BMPG	BMPO

- SCMD = MESPATB[18 : 16] (Secondary Command)
  パケットの SCMD フィールド。
- PCMD = MESPATB[15 : 12] (Primary Command)
  パケットの PCMD フィールド。
- ACC = MESPATB[11] (Accepted)
  パケットを受理した。
- DID = MESPATB[10 : 8] (Destination ID)
  パケットの DID フィールド。

- RET = MESPATB[7] (Retried)
  パケットがリトライさせられた。
- SID = MESPATB[6 : 4] (Source ID)
  パケットの SID フィールド。
- BMPH = MESPATB[3] (Backmap Hit)
  パケットはバックマップにヒットした。
- BMPV = MESPATB[2] (Backmap Valid)
  バックマップの V (Valid) ビット。
- BMPG = MESPATB[1] (Backmap Global)
  バックマップの G (Global) ビット。
- BMPO = MESPATB[0] (Backmap Owned)
  バックマップの O (Owned) ビット。

[初期値] MESCTRB = 0, 他は不定である。適切な値に初期化する。

MESPATP, MESMSKP, MESCTRP (`0x800001A0...B0`)

プロセッサから受けとった MCMD, その時点でのキャッシュ・ラインのステータス，並びにアドレス比較結果が MESPATP に等しければ，MESCTRP がインクリメントされる。なおパケット種別の比較の際に，MESMSKP によるマスクが施される。即ち MESPATP のビット i との比較結果を $e[i]$ とすると，$\prod(e[i]+\overline{\mathrm{MESMSKP}[i]})$ が真であれば MESCTRP がインクリメントされる。MESCTRP は 32 ビット幅であるので，約 86 秒分の CPU からの要求に関する集計を行なうことができる。MESPATP の各ビットの意味は以下の通り。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMMY	CCBL	L1IV	TAGH	TAGV	TAGX	TAGL	TAGD	SIZE		!RD	!WR	MCMD			

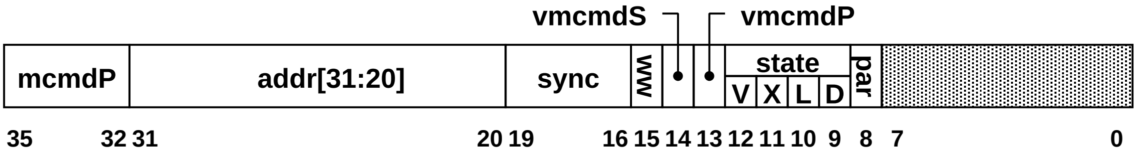
- DUMMY = MESPATP[15] (Dummy Page)
  ダミー・ページ・フラグ (PA[29]) の値。
- CCBL = MESPATP[14] (L1 Cachable)
  L1 cache にキャッシュ可であった。
- L1IV = MESPATP[13] (L1 Cache Invalidated)
  L1 cache に存在する（可能性のある）二つのブロックの少なくとも一方が無効化された。
- TAGH = MESPATP[12] (Tag Hit)
  キャッシュにヒットした。
- TAGV = MESPATP[11] (Tag Valid)
  キャッシュ・タグの V (Valid) ビット。
- TAGX = MESPATP[10] (Tag Exclusive)
  キャッシュ・タグの X (Exclusive) ビット。
- TAGL = MESPATP[9] (Tag Local)
  キャッシュ・タグの L (Local) ビット。
- TAGD = MESPATP[8] (Tag Dirty)
  キャッシュ・タグの D (Dirty) ビット。

- SIZE = MESPATP[7 : 6] (Access Size)
  アクセスのバイト幅(0:byte, 1:halfword, 2:word, 3:doubleword)を示す。
- !RD = MESPATP[5] (Read)
  0であればloadまたはswap系のアクセスであった。
- !WR = MESPATP[5] (Read)
  0であればstoreまたはswap系のアクセスであった。
- MCMD = MESPATP[3 : 0] (Memory Command)
  MCMD (PA[35:32])の値。

[初期値] 不定である。適切な値に初期化する。

FMP[n] (`0xA0000000`...`A00FFFE0`)

2次キャッシュ・タグのフロント・マップ。



- mcmd[3:0] = FMP[35 : 32] (Memory Command)
  同じラインが1次キャッシュにキャッシュされている場合，その物理アドレスの最上位4ビット(MCMD)。
- addr[31:20] = FMP[31 : 20] (Block Address)
  格納されているラインの物理アドレスのビット[31:20]。
- sync[3:0] = FMP[19 : 16] (Full/Empty)
  sync[i]は，アドレスのビット[4:3]がiであるようなdoublewordの同期ビット。
- ww = FMP[15] (Write Waiting)
  1であれば，ライン中のいずれかのdoublewordに対する**s-write**を待っているようなスレッドが存在する。ww = 1であるようなラインに対する**s-write**は，ラインの状態に関わらず必ずMMC/MBPLに通知される
- vmcmdS = FMP[14] (Validity of Secondary MCMD)
  1であれば，物理アドレスの最上位4ビット(MCMD)が，MCMD[3:1] = 011, MCMD[0] = FMP.mcmd[0] (**f-read**)であるようなラインが1次キャッシュに存在する可能性がある。
- vmcmdP = FMP[13] (Validity of Primary MCMD)
  1であれば，物理アドレスの最上位4ビット(MCMD)がmcmdであるようなラインが1次キャッシュに存在する可能性がある。
- state.V = FMP[12] (Valid)
  1であれば，ラインは有効である。
- state.X = FMP[11] (Exclusive)
  1であれば，ラインは共有されていない(排他的である)。
- state.L = FMP[10] (Local)
  1であれば，ラインは他のクラスタでは共有されていない。
- state.D = FMP[9] (Dirty)
  1であれば，ラインを置換する時にクラスタ・メモリへの書き戻しが必要である。
- par = FMP[8] (Parity)
  偶数パリティ。

[初期値] 不定である。state.V = 0 に初期化する。

RBMP[n] (`0xE0000000`... `E01FFFE0`)
任意の値を書くとバックマップが読出され，その結果が BMPRDR に格納される。

[初期値] 不定である。WBMP により V = 0 に初期化する。

WBMP[n](d) (`0xF0000000`... `FFFFFFE0`)
任意の値を書くと，バックマップにアドレスに対応する値が書込まれる。ADDR[27:0] は以下のように解釈される。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1111				BA								a[14:11]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a[10:0]											---	P	V	G	O

- BA = ADDR[27 : 20] (Block Address)
  バックマップのアドレス部に書込まれる値。
- a = ADDR[19 : 5] (Backmap Address)
  バックマップのアドレス。
- P = ADDR[3] (Parity)
  バックマップの偶数パリティに書込まれる値。
- V = ADDR[2] (Valid)
  バックマップの V (Valid) ビットに書込まれる値。
- G = ADDR[1] (Global)
  バックマップの G (Global) ビットに書込まれる値。
- O = ADDR[0] (Owned)
  バックマップの O (Owned) ビットに書込まれる値。

索引