

R800 ユーザーズマニュアル 暫定版

株式会社アスキー
システム事業部

1990年8月28日

本書は株式会社アスキーの開発した CPU、R800 のマニュアルです。
本書の内容の一部または全部を無断転載することは禁止されています。
本書の内容については、将来予告なく変更することがあります。
Copyright (c) 1990, ASCII Corporation

<注意>このデータ・シートの中には、ノンプルの無いページも含まれています(編集部)。

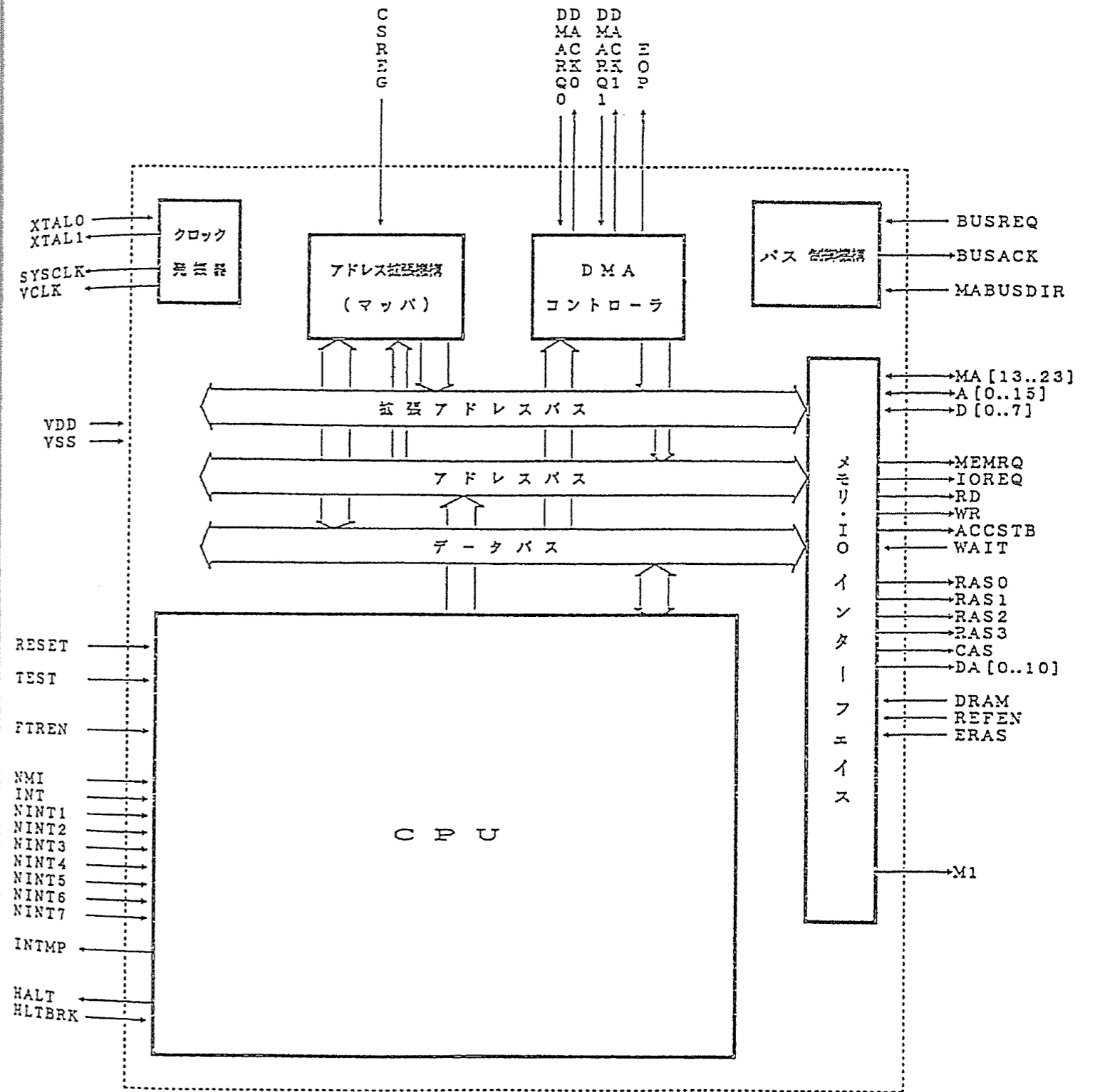
目次

1 R800 の特徴	1
2 ブロック図	2
3 ピン配置図	3
4 パッケージ寸法図	4
5 端子説明	5
5.1 端子表	5
5.2 端子機能説明	7
5.2.1 メモリ・I/O アクセス インターフェイス	7
5.2.2 割り込み インターフェイス	8
5.2.3 バス交換、DMA インターフェイス	9
5.2.4 クロック ジェネレータ	9
5.2.5 その他	10
6 電氣的仕様	11
6.1 絶対最大定格 (ABSOLUTE MAXIMUM RATINGS)	11
6.2 推奨動作条件 (RECOMMENDED OPERATING CONDITIONS)	11
6.3 推奨動作条件下での直流特性 (DC CHARACTERISTICS)	11
6.4 入出力端子容量 (INPUT / OUTPUT CAPACITY)	11
6.5 交流特性 (AC CHARACTERISTICS)	12
6.5.1 クロックタイミング	12
6.5.2 リセットタイミング	12
6.5.3 メモリ、I/O アクセス共通タイミング	13
6.5.4 External Address Mode	14
7 割り込み動作	15
8 DMA コントローラ	17
9 メモリマップ	18
10 DRAM インターフェース	21
11 リセットとクロック	22
12 内部拡張レジスタ	23
13 命令の実行	33
14 R800 の命令表	33

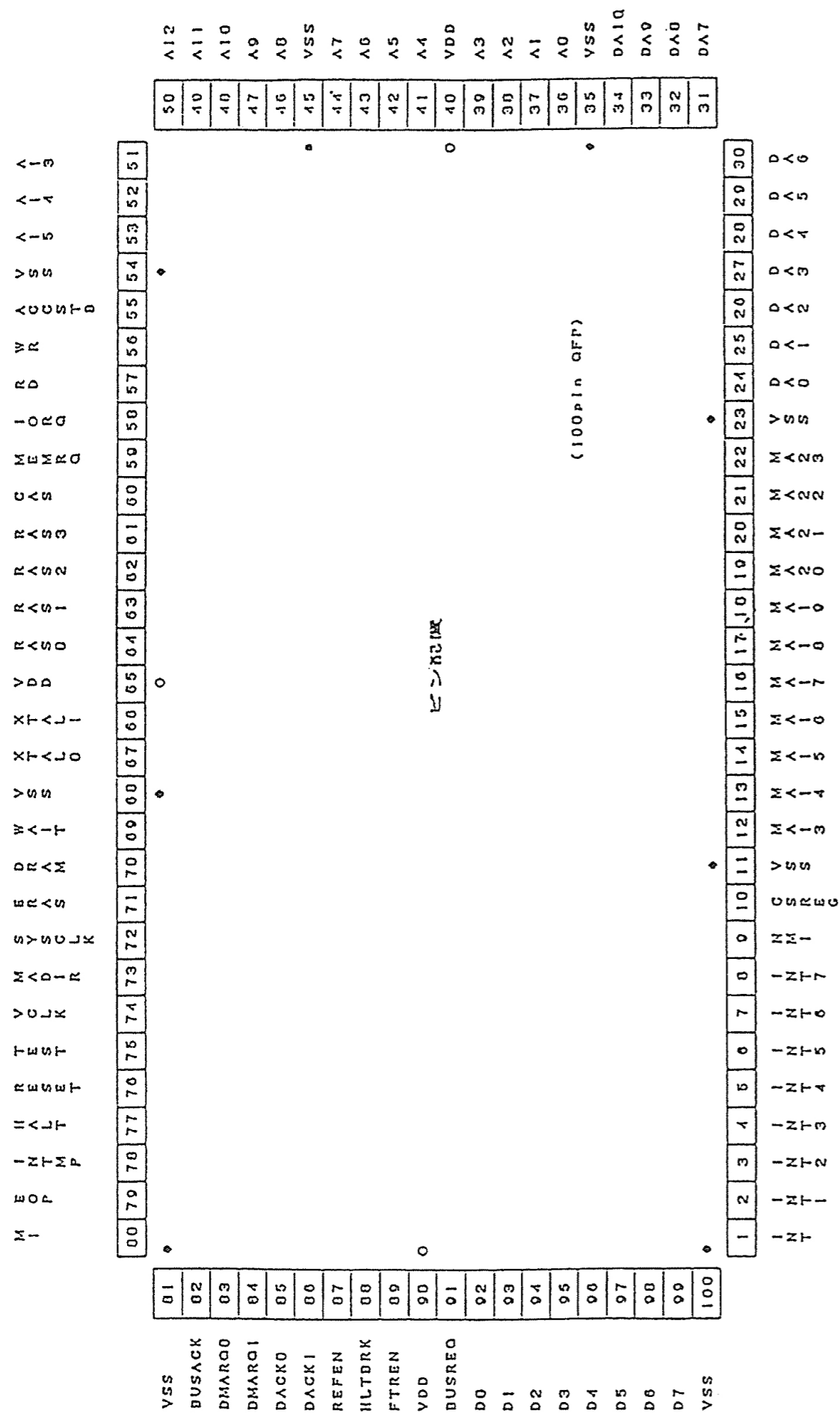
1 R800 の特徴

1. 16ビットALUバスを採用し演算処理を高速化している。
 - 16ビット算術演算、論理演算を1システムクロックで実行する。
2. 24ビット幅のアドレス空間である
 - メモリマップを9個内蔵し16MBまでメモリ拡張可能。
3. DRAMインターフェースを内蔵しているため、直接DRAMを接続できる。
 - 直接DRAMを接続できる
 - リフレッシュコントローラを内蔵している（リフレッシュはCAS BEFORE RAS方式）
 - DRAM高速インターフェース（ページモード）をサポートしているため、ノーウェイトでアクセス可能。
4. クロックジェネレータを内蔵。（28MHz）
5. インタラプト機能の強化。（7レベルの優先順位を持つ間接アドレッシングインタラプト）
6. DMAコントローラを2チャンネル内蔵。
7. CPUクロックは約7MHzで1バイト命令のほとんどを1CPUクロックで実行可能。
8. ザイログ社のZ80と命令コード上位コンパチブル。
 - Z80命令はすべてサポートしている
 - IX、IYレジスタの8ビット命令をサポートしている
 - 乗算命令をサポートしている
 - FASTモード（命令フェッチが1回のみでI/Oデータのブロック転送を行う）をサポートしている
9. 入出力命令の高速転送モード（命令のフェッチが1回のみ）をサポート。
10. マップアドレスピンを双方向化。
 - バス開放時、外部から全アドレスとERASを入力することにより、DRAMの全空間を外部からアクセスできる

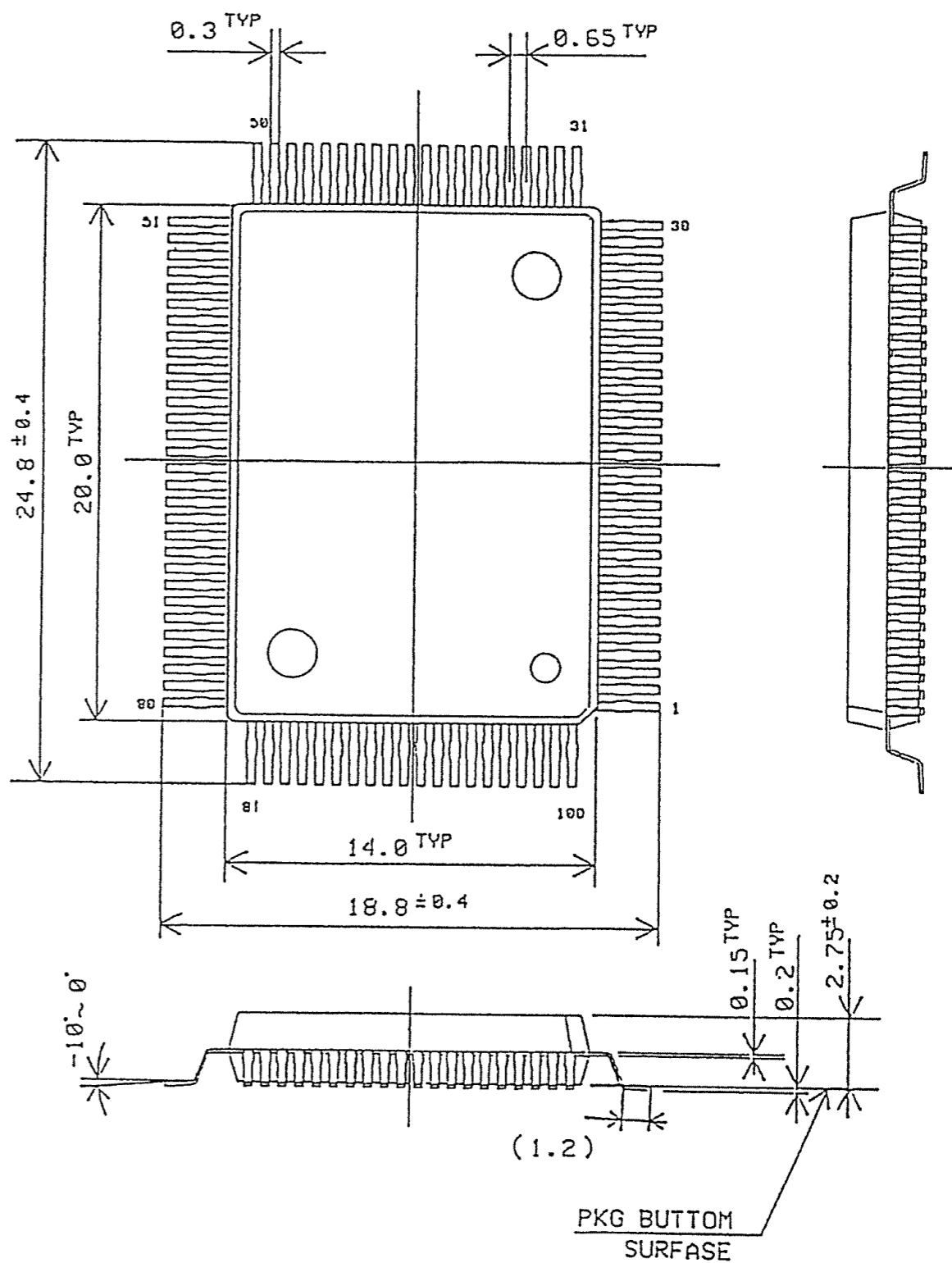
2 ブロック図



3 ピン配置図



4 パッケージ寸法図



5 端子説明

5.1 端子表

					*:Low active
100p	Pin Name	PinNo	I / O	state	Function
	*RESET	76	I	2	RESET
	XTAL0	67	I	2	XTAL input / MCK input (MCK=28.63636MHz)
	XTAL1	66	O	2	XTAL output
	SYSCLK	72	O	2	SYSCLK output (SYSCLK= 7.15909MHz) MCK / 4
	VCLK	74	O	2	VCLK output (VCLK =14.31818MHz) MCK / 2
	TEST1	75	I	2	test mode
	*M1	80	O	2	machine cycle 1
	*BUSREQ	91	I	2	BUS request
	*BUSACK	82	O	2	BUS acknowledge
	*TEST2	77	O	2	test pin
	*TEST3	88	I	2	test pin
12	FTREN	89	I	2	fast transfer enable
	*REFEN	87	I	2	refresh enable
	DRAM	70	I	2	DRAM select
	*MABUSDIR	73	I	2	select MA BUS direction
	*ERAS	71	I	2	external RAS
	*WAIT	69	I	2	wait request
	*MEMRQ	59	O	3	memory access request command
	*IORQ	58	O	3	io access request command
	*RD	57	O	3	read enable
	*WR	56	O	3	write enable
	*ACCSTB	55	O	3	access strobe
	*RAS0	64	O	2	(DRAM) RAS0(row address strobe)
	*RAS1	63	O	2	(DRAM) RAS1(row address strobe)
	*RAS2	62	O	2	(DRAM) RAS2(row address strobe)
	*RAS3	61	O	2	(DRAM) RAS3(row address strobe)
15	*CAS	60	O	2	(DRAM) CAS(column address strobe)
	*NMI	9	I	2	NMI(non maskable interrupt)
	*INT	1	I	2	interrupt
	*NINT1	2	I	2	new interrupt 1
	*NINT2	3	I	2	new interrupt 2
	*NINT3	4	I	2	new interrupt 3
	*NINT4	5	I	2	new interrupt 4
	*NINT5	6	I	2	new interrupt 5
	*NINT6	7	I	2	new interrupt 6
	*NINT7	8	I	2	new interrupt 7
10	*INTMP	78	O	2	enable MAP0', int sequence
	D7	99	I / O	3	data bus 7
	D6	98	I / O	3	data bus 6
	D5	97	I / O	3	data bus 5
	D4	96	I / O	3	data bus 4
	D3	95	I / O	3	data bus 3
	D2	94	I / O	3	data bus 2
	D1	93	I / O	3	data bus 1
8	D0	92	I / O	3	data bus 0

	A15	53	I / O	3	address bus 15, memory configuration data 1
	A14	52	I / O	3	address bus 14, memory configuration data 0
	A13	51	I / O	3	address bus 13
	A12	50	I / O	3	address bus 12
	A11	49	I / O	3	address bus 11
	A10	48	I / O	3	address bus 10
	A9	47	I / O	3	address bus 9
	A8	46	I / O	3	address bus 8
	A7	44	I / O	3	address bus 7
	A6	43	I / O	3	address bus 6
	A5	42	I / O	3	address bus 5
	A4	41	I / O	3	address bus 4
	A3	39	I / O	3	address bus 3
	A2	38	I / O	3	address bus 2
	A1	37	I / O	3	address bus 1
16	A0	36	I / O	3	address bus 0
	MA23	22	I / O	3	mapper address bus 23
	MA22	21	I / O	3	mapper address bus 22
	MA21	20	I / O	3	mapper address bus 21
	MA20	19	I / O	3	mapper address bus 20
	MA19	18	I / O	3	mapper address bus 19
	MA18	17	I / O	3	mapper address bus 18
	MA17	16	I / O	3	mapper address bus 17
	MA16	15	I / O	3	mapper address bus 16
	MA15	14	I / O	3	mapper address bus 15
11	MA14	13	I / O	3	mapper address bus 14
	MA13	12	I / O	3	mapper address bus 13
	DA10	34	O	2	DRAM address bus MA20 / MA21
	DA9	33	O	2	DRAM address bus MA18 / MA19
	DA8	32	O	2	DRAM address bus MA16 / MA17
	DA7	31	O	2	DRAM address bus A7 / MA15
	DA6	30	O	2	DRAM address bus A6 / MA14
	DA5	29	O	2	DRAM address bus A5 / MA13
	DA4	28	O	2	DRAM address bus A4 / A12
	DA3	27	O	2	DRAM address bus A3 / A11
	DA2	26	O	2	DRAM address bus A2 / A10
	DA1	25	O	2	DRAM address bus A1 / A9
11	DA0	24	O	2	DRAM address bus A0 / A8
	*CSREG	10	I	2	chip select of I / O address / data register
	*DMARQ0	83	I	2	DMA request # 0
	*DMARQ1	84	I	2	DMA request # 1
	*DACK0	85	O	2	DMA acknowledge # 0
	*DACK1	86	O	2	DMA acknowledge # 1
6	*EOP	79	O	2	DMA end of process
	VDD	40	I	-	+5V
	VDD	65	I	-	+5V
	VDD	90	I	-	+5V
	VSS	11	I	-	GND
	VSS	23	I	-	GND
	VSS	35	I	-	GND
	VSS	45	I	-	GND
	VSS	54	I	-	GND
	VSS	68	I	-	GND
	VSS	81	I	-	GND
11	VSS	100	I	-	GND

5.2 端子機能説明

5.2.1 メモリ・I/O アクセス インターフェイス

D7-0 (I/O, 3 state) 8ビット双方向データバス。

A15-0 (I/O, 3 state) 16ビットアドレスバス。

A15-14 は R800 に接続される DRAM のタイプを示す値として RESET が Low から High に立ち上がったときに読み込まれる。

config data 1 (=A15)	config data 0 (=A14)	使用する DRAM type
Low	Low	64k × 4bit
Low	High	256k × 4bit
High	Low	1M × 4bit
High	High	4M × 4bit

入力と出力の切り替え

次のいずれかの場合、入力状態になる。

1. BUSACK = Low のとき
2. RESET = Low のとき

MA23-13 (I/O, 3 state) 11ビット拡張アドレスバス。

入力と出力の切り替え

次のいずれかの場合、MA23-13 の値が入力される。

1. MABUSDIR = Low のとき (入力)
2. RESET = Low のとき (入力)
3. BUSACK = Low のとき (ハイインピーダンス)

DA10-0 (O, 2 state) DRAM 専用の 11ビットアドレスバス。

DRAM (I) DRAM をアクセスしていることを R800 に伝える入力信号。この信号はメモリアクセス毎にダイナミックに変化する。この信号をもとに R800 は RAS / CAS のコントロールを行う。

* REFEN (I) この信号がアクティブの場合、R800 はリフレッシュを行う。

* MABUSDIR (I) マップの選択を指定する。

1: R800 内のマップを使う
(この時 MA23-MA11 は出力としてはたらく)

0: R800 外のマップを使う
(この時 MA23-MA11 は入力としてはたらく)

* WAIT (I) メモリまたは I/O から R800 に対するウェイト要求。

* MEMRQ (O, 3 state) メモリアクセス要求モードを示す出力信号。
(正確には「I/O アクセス要求モード以外」であることを示す)

RESET = Low または BUSACK = Low のときハイインピーダンス状態となる。

* IORQ (O, 3 state) I/O アクセス要求モードを示す出力信号。

RESET = Low または BUSACK = Low のときハイインピーダンス状態となる。

* RD (O, 3 state) メモリまたは I/O に対して READ 要求モードであることを示す出力信号。

(正確には「WRITE 要求モード以外」であることを示す)
RESET = Low または BUSACK = Low のときハイインピーダンス状態となる。

* WR (O, 3 state) メモリまたは I/O に対して WRITE 要求モードであることを示す出力信号。

RESET = Low または BUSACK = Low のときハイインピーダンス状態となる。

* ACCSTB (O, 3 state) メモリアクセスまたは I/O アクセス時のアクセスストロブ出力信号。

RESET = Low または BUSACK = Low のときハイインピーダンス状態となる。

* RAS0-3 (O, 2 state) DRAM に対する RAS 信号。

RESET = Low または BUSACK = Low のときハイインピーダンス状態となる。

* CAS (O, 2 state) DRAM に対する CAS 信号。

* ERAS (I) BUSACK = 0、REFEN = 1 の時、R800 は ERAS の信号をそのまま RAS0-RAS3 へ出力する。

5.2.2 割り込み インターフェイス

* NMI (I) ノンマスクابلな (割り込みマスクできない) 割り込み信号。old mode の場合も new mode の場合も使用する。

* INT (I) 割り込み信号 (OLD MODE)。

* NINT1-7 (I) INT とは別の割り込み信号 (NEW MODE)。

* INTMP (O) OLD INT の場合: 割り込みが受け付けられて、R800 の割り込みシーケンスが実行中であることを示す。

NEW INT の場合: MAP0' 使用中を示す。すなわち、NMI または NINT1-7 が受け付けられてから RETN または RETI が実行されるまでの期間を示す。

(注意)

1. INT と NINT1-7 は同時には使えない。

RESET 後は INT が有効で NINT1-7 の信号は無視される。その後ソフトウェアにより、R800 内部のレジスタである "int upper register" に値がロードされると、それ以降は、NINT1-7 が有効で INT の信号は無視される。

2. NINT1-7 の中で優先順位は、NINT1 が最も高く NINT7 が最も低い。

5.2.3 バス交換、DMA インターフェイス

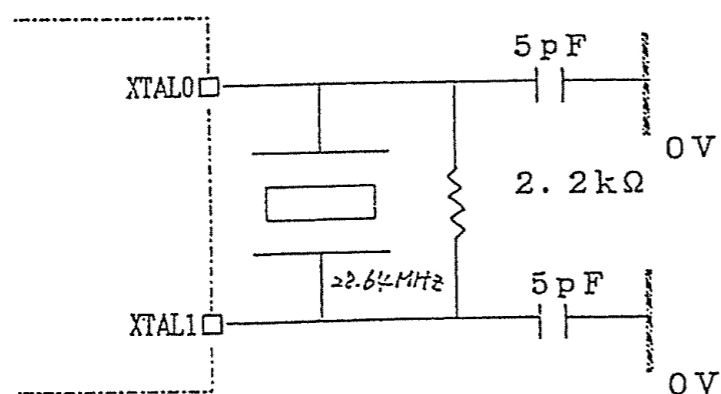
- * BUSREQ (I) R800 以外のバスマスタがメモリまたは I/O をアクセスしたい場合、R800 に対してバス開放を要求する信号。
- * BUSACK (O) バス開放許可信号。この信号がアクティブの場合は、A15-0、MA23-13、MEMRQ、IORQ、RD、WR、ACCSTB の出力信号はすべてハイ・インピーダンス状態となる。
- * DMARQ0 (I) R800 内部の DMA コントローラ 0 に対する、DMA 転送要求信号。
- * DMARQ1 (I) R800 内部の DMA コントローラ 1 に対する、DMA 転送要求信号。
- * DACK0 (O) DMARQ0 を受け付け、DMA コントローラ 0 が実行中であることを示す。
- * DACK1 (O) DMARQ1 を受け付け、DMA コントローラ 1 が実行中であることを示す。
- * EOP (O) 指定バイト数の DMA 転送が終了したことを示す。
DMA0、DMA1 のどちらが終了したかは内部レジスタ DMA0_mode_reg または DMA1_mode_reg を読んで判断すること。

(注意)

1. SYSCLK に同期した信号である。
BUSACK、DACK0、DACK1、EOP は SYSCLK に同期して出力される。
BUSREQ、DMARQ0、DMARQ1 は SYSCLK に同期させる必要はない。
2. DMA0 は DMA1 より優先度が高い。
DACK0 と DACK1 が両方ともアクティブになることはない。
3. DMA 転送中は CPU は CPU インストラクションを実行しない。

5.2.4 クロック ジェネレータ

- XTAL0 (I) 28.63636MHz の水晶振動子の接続端子。
- XTAL1 (O) 外部で発振させたクロックを入力する時は XTAL0 を用いる。
この場合は 50% duty の波形を入れること。
このクロックを MCK (main clock) と呼ぶ。



(水晶振動子の接続例)

・ R、C1、C2 の値

タイプ 1 R = 500K Ω ~ 2M Ω、C1 = C2 = 12pF ~ 33pF

タイプ 2 R = 2.4K Ω ~ 5.6K Ω、C1 = C2 = 5pF ~ 18pF

タイプ 1 の水晶振動子例：UM-1 (MF 型) 28.63636MHz [朝日電波 (株)]

タイプ 2 の水晶振動子が多い。

補足：

発振周波数が高くなるとオーバートーンの水晶振動子が多くなる。

オーバートーンの水晶振動子は各メーカーにより R、C の値が異なる。

中には、R の大きい (1M Ω レベル) ところで安定するものもあるし、また、中には、R の小さい (数 K Ω レベル) ところで発振し、R が大きくなると基本波で発振するものもある。

SYSCLK (O) MCK を 4 分周したもの (SYSCLK = 7.15909MHz)

VCLK (O) MCK を 2 分周したもの (VCLK = 14.31818MHz)

5.2.5 その他

- * RESET (I) アクティブ (Low) 時に R800 が初期化される。R800 内部で RESET される (値が 0 になる) レジスタ、フリップフロップは IR、PC、IFF1、IFF2、R、I、MAP0 である。この信号がアクティブの場合は、A15-0、MA23-13、MEMRQ、IORQ、RD、WR、ACCSTB の出力信号はすべてハイ・インピーダンス状態となる。

(注)

1. RESET 直後のリフレッシュについて。
R800 は RESET 直後 DRAM に対するリフレッシュを行わない。従ってプログラムで一定時間 (4ms 以上) ダミールーチンを行わせること。
2. MCK の 4 クロック分以下の Low パルスは RESET 信号として認められない。

TEST (I) 一般利用者は常に Low にしておくこと。

* M1 (O) マシンサイクル 1 を示す。

* TEST2 (O) テスト端子 (OTHER 出力)。

* TEST3 (I) テスト端子 (halt 状態を解除するための要求信号)、通常 High にしておくこと。

(注)

halt 状態における R800 の内部状態

割り込み、DMARQ0、DMARQ1、BUSREQ は受け付けられる
リフレッシュは行い (但し REFEN = Low の場合)

- * CSREG (I) R800 内の I/O アドレスレジスタまたは I/O データレジスタを選択する。
もし A0 = 0 なら I/O アドレスレジスタを選択する。
もし A0 = 1 なら I/O データレジスタを選択する。

VDD 電源入力 (+5V)

VSS 電源入力 (0V)

6 電気的仕様

6.1 絶対最大定格 (ABSOLUTE MAXIMUM RATINGS)

(VSS = 0.0V 基準)

Characteristic	Symbol	Value		Unit
		min	max	
Supply voltage	VDD	-0.3	+7.0	V
Input voltage	VI	-0.3	VDD + 0.3	V
Output voltage	VO	-0.3	VDD + 0.3	V
Storage temperature	Tstg	-50	+125	°C
Operating temperature	Top	0	70	°C

6.2 推奨動作条件 (RECOMMENDED OPERATING CONDITIONS)

(VSS = 0.0V 基準)

Characteristic	Symbol	Value			Unit
		min	typ	max	
Supply voltage	VDD	4.75	5.0	5.25	V
Operating temperature	Top	0	25	70	°C
High level input voltage (group 1)	VIH1	3.5	-	-	V
Low level input voltage (group 1)	VIL1	-	-	1.0	V
High level input voltage (group 2)	VIH2	2.2	-	-	V
Low level input voltage (group 2)	VIL2	-	-	0.8	V

group1: XTAL0 (CMOS 入力インターフェイス)

group2: group 1 以外の入力端子 (TTL 入力インターフェイス)

6.3 推奨動作条件下での直流特性 (DC CHARACTERISTICS)

(VDD = 5 ± 0.25V、TA = 0~70 °C)

Characteristic	Symbol	Rating		Unit
		min	max	
High level output voltage	VOH (IOH=-2.0mA)	VDD - 1.0	-	V
Low level output voltage	VOL (IOL= 2.0mA)	-	0.4	V
Input leakage current	IL	-10	+10	μA
Power supply current	IDD	-	55	mA

6.4 入出力端子容量 (INPUT / OUTPUT CAPACITY)

測定条件: Ta = 25.0 °C VDD = 5.00V

入力電圧: VIH = 2.4V 以上 VIL = 0.8V 以下

入力信号周波数: 1.0MHz

VDD と測定端子以外は GND に接続して測定

Characteristic	Symbol	Rating			Unit
		min	typ	max	
Input pin	Ci	-	5.0	8.0	pF
Output pin	Co	-	-	10.0	pF
Input / output pin	Cio	-	-	12.0	pF

6.5 交流特性 (AC CHARACTERISTICS)

測定条件: Ta = 0~70 °C VDD = 4.75V~5.25V

入力電圧: VIH = 2.4V VIL = 0.45V

入力パルス立ち上がり時間 (VIL から VIH): 5nS 以内

入力パルス立ち下がり時間 (VIH から VIL): 5nS 以内

タイミング測定電圧: VOL = 0.8V VIL = 0.8V

VOH = 3.5V VIH = 2.0V

6.5.1 クロックタイミング

測定条件: CL=50pf、単位: ns

番号	記号	項目	min	typ	max
1	TcX	XTAL Cycle Time (注1)	34.9		
2	TwXh	XTAL Pulse Width (High)	10		
3	TwXl	XTAL Pulse Width (Low)	10		
4	TfX	XTAL Fall Time Delay		1	5
5	TrX	XTAL Rise Time Delay		1	5
6	TdX(VCLKr)	XTAL ↑ to VCLK ↑ Delay			20
7	TdX(VCLKf)	XTAL ↑ to VCLK ↓ Delay			20
8	TdX(SYSCLKr)	XTAL ↑ to SYSCLK ↑ Delay			20
9	TdX(SYSCLKf)	XTAL ↑ to SYSCLK ↓ Delay			20

fig1 参照

(注1) XTAL の最大周波数は、fmax=28.63636MHz

6.5.2 リセットタイミング

測定条件: CL=50pf、単位: ns

番号	記号	項目	min	typ	max
10	TwRESETl	RESET Pulse Width (Low)	4 · TcX		

fig2 参照

6.5.3 メモリ、I/O アクセス共通タイミング

測定条件: CL=50pf, 単位: ns

番号	記号	項目 (注1)	PH	min	typ	max
1	TdX(MEMRQf)	XTAL0 ↑ to MEMRQ ↓ Delay	PH1			45
2	TdX(MEMRQr)	XTAL0 ↑ to MEMRQ ↑ Delay	PH0			45
3	TdX(IORQf)	XTAL0 ↑ to IORQ ↓ Delay	PH1			45
4	TdX(IORQr)	XTAL0 ↑ to IORQ ↑ Delay	PH0			45
5	TdX(RDf)	XTAL0 ↑ to RD ↓ Delay	PH1			45
6	TdX(RDr)	XTAL0 ↑ to RD ↑ Delay	PH0			45
7	TdX(WRf)	XTAL0 ↑ to WR ↓ Delay	PH1			45
8	TdX(WRr)	XTAL0 ↑ to WR ↑ Delay	PH0			45
9	TdX(ACCSTBf)	XTAL0 ↑ to ACCSTB ↓ Delay	PH2			25
10	TdX(ACCSTBr)	XTAL0 ↑ to ACCSTB ↑ Delay	PH0			25
11	TdX(RASf)	XTAL0 ↑ to RAS ↓ Delay	PH0			35
12	TdX(RASr)	XTAL0 ↑ to RAS ↑ Delay	PH1			55
13	TdX(R・CASf)	XTAL0 ↑ to CAS ↓ Delay (read)	PH1			50
14	TdX(R・CASr)	XTAL0 ↑ to CAS ↑ Delay (read)	PH0			25
15	TdX(W・CASf)	XTAL0 ↑ to CAS ↓ Delay (write)	PH2			30
16	TdX(W・CASr)	XTAL0 ↑ to CAS ↑ Delay (write)	PH0			25
17	TdX(REF・CASf)	XTAL0 ↑ to CAS ↓ Delay (refresh)	PH2			30
18	TdX(REF・CASr)	XTAL0 ↑ to CAS ↑ Delay (refresh)	PH2			25
20	TdX(A)	XTAL0 ↑ to Address Valid Delay	PH1			35
21	TdX(Ah)	XTAL0 ↑ to Address Valid Delay (hold)	PH1	10		
24	TdX(D) [output]	XTAL0 ↑ to Data Valid Delay	PH1	10		45
25	TdX(Dh) [output]	XTAL0 ↑ to Data Valid Delay (hold)	PH1	10		
26	TsD(X) [input]	Data Setup Time to XTAL0 ↑ IN 命令の時 (注2) 上記以外の時	PH0	25		
27	ThD(X) [input]	Data Hold Time to XTAL0 ↑	PH0	10		
30	TdRASf(DA)	DRAM Address Stable prior to RAS ↓	-	30		
31	ThDA(RASf)	DRAM Address Hold Time to RAS ↓	-	15		
32	TdR・CASf(DA)	DRAM Address Stable prior to CAS ↓ (Read)	-	5		
33	ThDA(R・CASf)	DRAM Address Hold Time to CAS ↓ (Read)	-	30		
34	TdW・CASf(DA)	DRAM Address Stable prior to CAS ↓ (Write)	-	10		
35	ThDA(W・CASf)	DRAM Address Hold Time to CAS ↓ (Write)	-	30		
36	TdCASf(D)	Data Stable prior to CAS ↓ (Write)	-	10		
37	TdD(CASf)	Data Hold Time to CAS ↓ (Write)	-	70		
38	TPR(RAS)	RAS pre charge time	-	80		
39	TPR(R・CAS)	CAS pre charge time (page mode, Read)	-	25		
40	TPR(W・CAS)	CAS pre charge time (page mode, Write)	-	50		
41	TdMA(DA)	MA to DRAM Address Valid Delay (Turbo Mode)	-			25

fig3-1~fig3-7 参照

注1: PHとはPHASEの略です。

SYSCLKの1サイクルをXTAL0毎に4つのPHASEに分け、それぞれPH0、PH1、PH2、PH3と記述してあります。

注2: 次のIN命令の時

IN r、(C)
IN M、(C)

測定条件: CL=50pf, 単位: ns

番号	記号	項目	PH	min	typ	max
50	TsWAIT(X)	WAIT Setup Time to XTAL0 ↑	PH0	25		
51	ThWAIT(X)	WAIT Hold Time to XTAL0 ↑	PH0	10		
52	TsDRAM	DRAM Setup Time to XTAL0 ↑	PH0	25		
53	ThDRAM	DRAM Hold Time to XTAL0 ↑	PH0	10		
54	TdX(M1f)	XTAL0 ↑ to M1 ↓ Delay	PH0			70
55	TdX(M1r)	XTAL0 ↑ to M1 ↑ Delay	PH0			70
60	TsBUSREQ(X)	BUSREQ Setup Time to XTAL0 ↑ (注1)	PH0	30		
61	ThBUSREQ(X)	BUSREQ Hold Time to XTAL0 ↑	PH0	10		
62	TdX(BUSACKf)	XTAL0 ↑ to BUSACK ↓ Delay	PH0			35
63	TdX(BUSACKr)	XTAL0 ↑ to BUSACK ↑ Delay	PH0			35
64	TdX(Az)	XTAL0 ↑ to Address Float Delay	PH0			35
66	TdX(Dz)	XTAL0 ↑ to DATA Float Delay	PH0			35
67	TdX(CTz)	XTAL0 ↑ to Control Output Float Delay	PH0			35
68	TdX(INTMPf)	XTAL0 ↑ to INTMP ↓ Delay (OLDINTの場合)	PH0			55
69	TdX(INTMPr)	XTAL0 ↑ to INTMP ↑ Delay (OLDINTの場合)	PH0			45
70	TwNMI	NMI Pulse Width	-	30		
71	TsINT(X)	INT Setup Time to XTAL0 ↑ (注1)	PH0	30		
72	TdX(TEST2 ↑)	XTAL0 ↑ to TEST2 (OTHER) ↑ Delay	PH0			45
73	Tdx(TEST2 ↓)	XTAL0 ↑ to TEST2 (OTHER) ↓ Delay	PH0			45
83	TsFTREN(X)	FTREN Setup Time to XTAL0 ↑ (注1)	PH0	30		
84	TsREFEN(X)	REFEN Setup Time to XTAL0 ↑ (注1)	PH0	30		
85	TdC(MEMRQr)	SYSCLK ↑ to MEMRQ ↑ delay	-	10		30
86	TdC(IORQr)	SYSCLK ↑ to IORQ ↑ delay	-	10		30
87	TdC(RDr)	SYSCLK ↑ to RD ↑ delay	-	10		30
88	TdC(WRr)	SYSCLK ↑ to WR ↑ delay	-	10		30
89	TdC(ACCSTBr)	SYSCLK ↑ to ACCSTB ↑ delay	-	0		10
90	TsA(ACCSTBf)	Address Setup Time to ACCSTB ↓	-	10		
91	ThA(ACCSTBr)	Address Hold Time to ACCSTB ↑	-	20		
92	TsD(ACCSTBf)	Data Setup Time to ACCSTB ↓	-	10		
93	ThD(ACCSTBr)	Data Hold Time to ACCSTB ↑	-	20		
94	TdACCSTBf(MEMRQf)	MEMRQ ↓ Stable prior to ACCSTB ↓	-	10		25
95	TdACCSTBf(IORQf)	IORQ ↓ Stable prior to ACCSTB ↓	-	10		25
96	TdACCSTBf(RDf)	RD ↓ Stable prior to ACCSTB ↓	-	10		25
97	TdACCSTBf(WRf)	WR ↓ Stable prior to ACCSTB ↓	-	10		25
98	TdDout(ACCSTBr)	ACCSTB ↑ to Data output mode delay (注2)	-	25		
99	TdWAIT(ACCSTBf)	ACCSTB ↓ to WAIT input	-	30		
100	TdDRAM(ACCSTBf)	ACCSTB ↓ to DRAM input	-	30		
101	TdACCSTBf(RASf)	ACCSTB ↓ to RAS ↓ delay	-			90
102	TdACCSTBf(R・CASf)	ACCSTB ↓ to CAS ↓ delay (read)	-			135
103	TdACCSTBf(W・CASf)	ACCSTB ↓ to CAS ↓ delay (write)	-			155

fig3-1~fig3-7 参照

(注1) もし、Setup Timeが守られない場合は、次のSYSCLKサイクルでR800内部のフリップフロップにラッチされる。

(注2) R800のD0-D7端子が入力から出力に切り替わるまでの時間。

[補足 番号85~103における値は、clock周波数が最大(fmax = 28.63636MHz)のときである]

(注1) もし、Setup Timeが守られない場合は、次のSYSCLK上のサイクルでR800内部のフリップフロップにラッチされる。

6.5.4 External Address Mode

入力条件 : /BUSACK = Low and /REFEN = High

出力 : 次の出力信号は、常にハイインピーダンスとなる。

/MEMRQ、/IORQ、/RD、/WR、/ACCSTB、D0-D7

測定条件：CL=50pf、単位：ns

番号	記号	項目	PH	min	typ	max
1	TdX(SYSCLK)	XTAL0 ↑ to SYSCLK ↑ Delay	-			20
2	TdA(DA)	A、MA to DRAM Address Delay (ROW ADD)	-			25
3	TsERAS(X)	ERAS Setup Time to XTAL0 ↑ (注1)	-	25		
4	TdX(DA・COL)	XTAL0 ↑ to DRAM Address Delay (COL ADD)	-			25
5	TdX(DA・ROW)	XTAL0 ↑ to DRAM Address Delay (ROW ADD)	-	5		
6	TdX(CASf)	XTAL0 ↑ to CAS ↓ Delay	-			20
7	TdERASf(RASf)	ERAS ↓ to RAS ↓ Delay	-			30
8	TdERASr(RASr)	ERAS ↑ to RAS ↑ Delay	-			25
9	TdERASr(CASr)	ERAS ↑ to CAS ↑ Delay	-			25
10	TdERASr(DA)	ERAS ↑ to DRAM Address Delay	-	25		

fig4 参照

(注1) もし、この条件が守られないときは、次サイクルの XTAL ↑ で受けつけられる。

(注2) TcX : XTAL Cycle Time

7 割り込み動作

R800 のインタラプトには、大きく分けて次の5つのモードがあります。

1. インタラプトモード0

Z80 のインタラプトモード0 とコンパチブル

2. インタラプトモード1

Z80 のインタラプトモード1 とコンパチブル

3. インタラプトモード2

Z80 のインタラプトモード2 とコンパチブル

4. NMI

Z80 のNMI とコンパチブル

5. 新インタラプトモード

R800 独自の割り込みモードで、NINT1 から7 と NMI の信号で割り込みを受け付けます。

インタラプトベクトルレジスタに設定された値が、ベクターテーブルの先頭上位8ビットになり、先頭下位8ビットはオール0です。

あらかじめベクターテーブルに用意されたデータを、NINT1~7、NMI によりセレクトしそのデータが割り込みルーチン開始アドレスとなります。

ADDRESS

XX00H	NMI	SERVICE	VECTOR
XX02H	NINT1	SERVICE	VECTOR
XX04H	NINT2	SERVICE	VECTOR
XX06H	NINT3	SERVICE	VECTOR
XX08H	NINT4	SERVICE	VECTOR
XX0AH	NINT5	SERVICE	VECTOR
XX0CH	NINT6	SERVICE	VECTOR
XX0EH	NINT7	SERVICE	VECTOR

INT VECTOR REG →

Z80 互換の割り込みモードから新インタラプトの切り替えはインタラプトベクトルレジスタを設定することで行なうことが出来ます。新インタラプトでは従来の INT ピンは無効になり、NMI ピンによる割り込み動作は、新インタラプト動作になります。NINT1 から7までのインタラプトはインタラプトマスクレジスタでマスクすることが出来ます。NMI と NINT1 から7それぞれのインタラプトには優先順位が有り、NMI が最優先で NINT7 が一番低くなっています。また割り込み受け付け時に PC の値は0番地と1番地に格納されます。割り込み処理中(インタラプトを受け付けてから RETI 命令または RETN 命令を実行するまで)にアドレスの A13 から A15 がオール0の時、マップは MAP0' が用いられます(システムソフトと、アプリケーションを分けることが出来る)。

インタラプト動作順序

メインルーチンを実行

↓

インタラプト受け付け

↓

マップ0' をセレクト

↓

プログラムカウンタを

0番地と1番地に格納

↓

サービスベクターの読み込み

(INT VECTOR REG × 100H + インタラプト番号 × 2) → PCL

(INT VECTOR REG × 100H + インタラプト番号 × 2 + 1) → PCH

↓

サービスルーチンを実行

↓

RETI または RETN 命令フェッチ

↓

プログラムカウンタの復帰

(0番地) → PCL

(1番地) → PCH

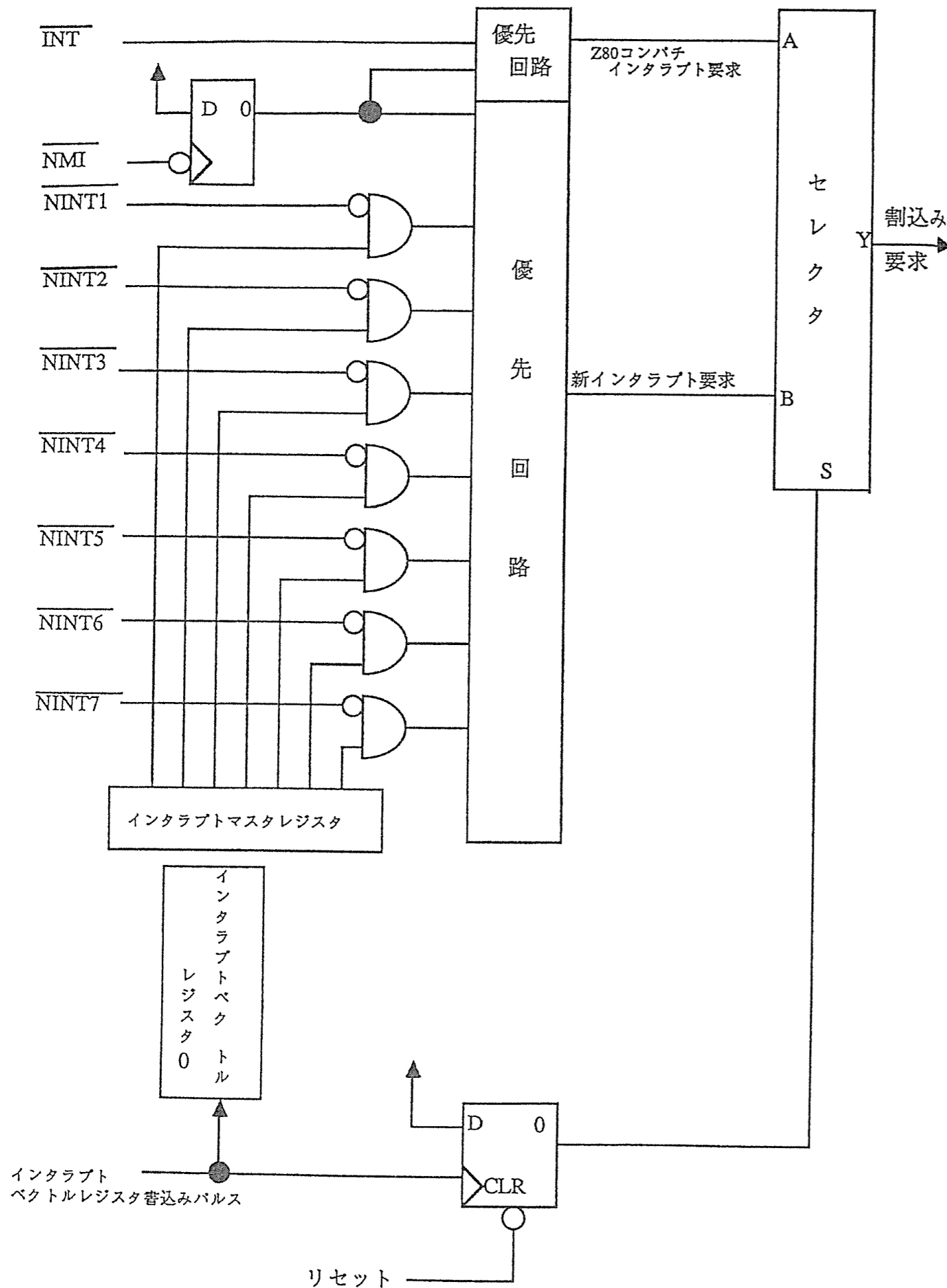
↓

マップ0-7 をセレクト

↓

メインルーチンに戻る

インタラプト部ブロック図



8 DMA コントローラ

R800ではDMAコントローラがDMA0とDMA1の2チャンネル内蔵されています。DMA0とDMA1ではDMA0の方が優先順位が高くなっています。転送先、転送元のアドレスはそれぞれ24ビット、転送バイト数は16ビットで設定可能です。カウンタは24ビットカウンタを用いているため64Kバイトのバウンダリはありません。

DMAモードレジスタにより以下の設定が出来ます。

1. 転送先、転送元のデバイスを選択可能です。(I/OまたはMEMORY)
2. 転送先、転送元のアドレスの自動インクリメント
3. DMA転送のイネーブル、ディセーブル
4. 転送の繰返し
5. 転送終了のフラグ読み込み

可能なDMAの種類

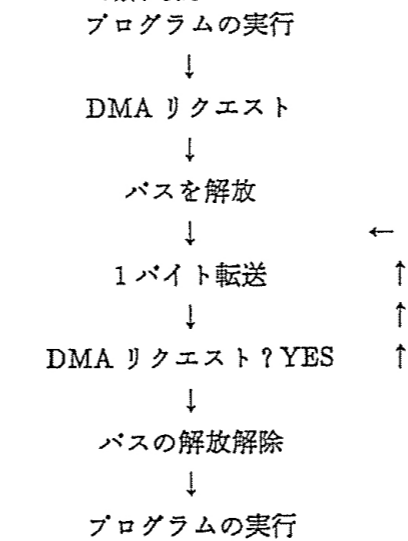
- メモリからメモリへの転送
- I/Oからメモリへの転送
- メモリからI/Oへの転送
- I/OからI/Oへの転送

転送の繰返しビット

このビットを「1」にすることで転送を繰り返して行ないます。D/Aコンバータに音を出力したりするとき便利です。カウンタの値が転送バイト数と等しくなったとき転送先アドレスと転送元アドレスがリロードされます。

DMAはDMARQピンを「LOW」にすることにより開始します。DMA転送の終了はEOPピンにも出力されます。

DMA動作順序



9 メモリマップ

R800 は最大 16Mb までメモリ空間を拡張できるメモリマップを内蔵しています。マップレジスタは各 11 ビットのレジスタで合計 9 本あり、8 本はアドレス信号 A13 から A15 でセレクトされ残りの 1 本は新インタラプト時 A13 から A15 がオール 0 の時セレクトされます。マップアドレスは MA13 から MA23 のピンに出力されます。また MA13 から MA23 のピンは双方向で MADIR 信号でコントロールでき、外部のバンクレジスタの出力をマップアドレスとして使用することも可能です。

バス解放要求

R800 は BUSRQ 信号によりデータバスをハイインピーダンスにすることが出来ます。この時ハイインピーダンスとなる信号は次の通りです。

- D0-D7
- RD、WR、IORQ、MEMRQ

またこの時次の信号は入力になり、R800 に直接接続された DRAM をアクセスすることが出来ます。

- A0-A15

外部からの DRAM アクセスの方法

R800 ではバス解放時 (BUSRQ=L, MADIR=L) にアドレスを入力することにより、直接接続された DRAM を外部のメモリリクエスト信号を ERAS ピンに接続してアクセスすることが出来ます。ERAS ピンを LOW レベルにすることにより、R800 内部でロウアドレスとコラムアドレスの切り替え及び CAS 信号の出力を行ないます。また RAS0 から RAS3 はリセット時に A14 と A15 ピンから読み込んだ DRAM の種類と外部から入力するアドレスの値によりセレクトされます。

高速 I/O アクセスモードについて

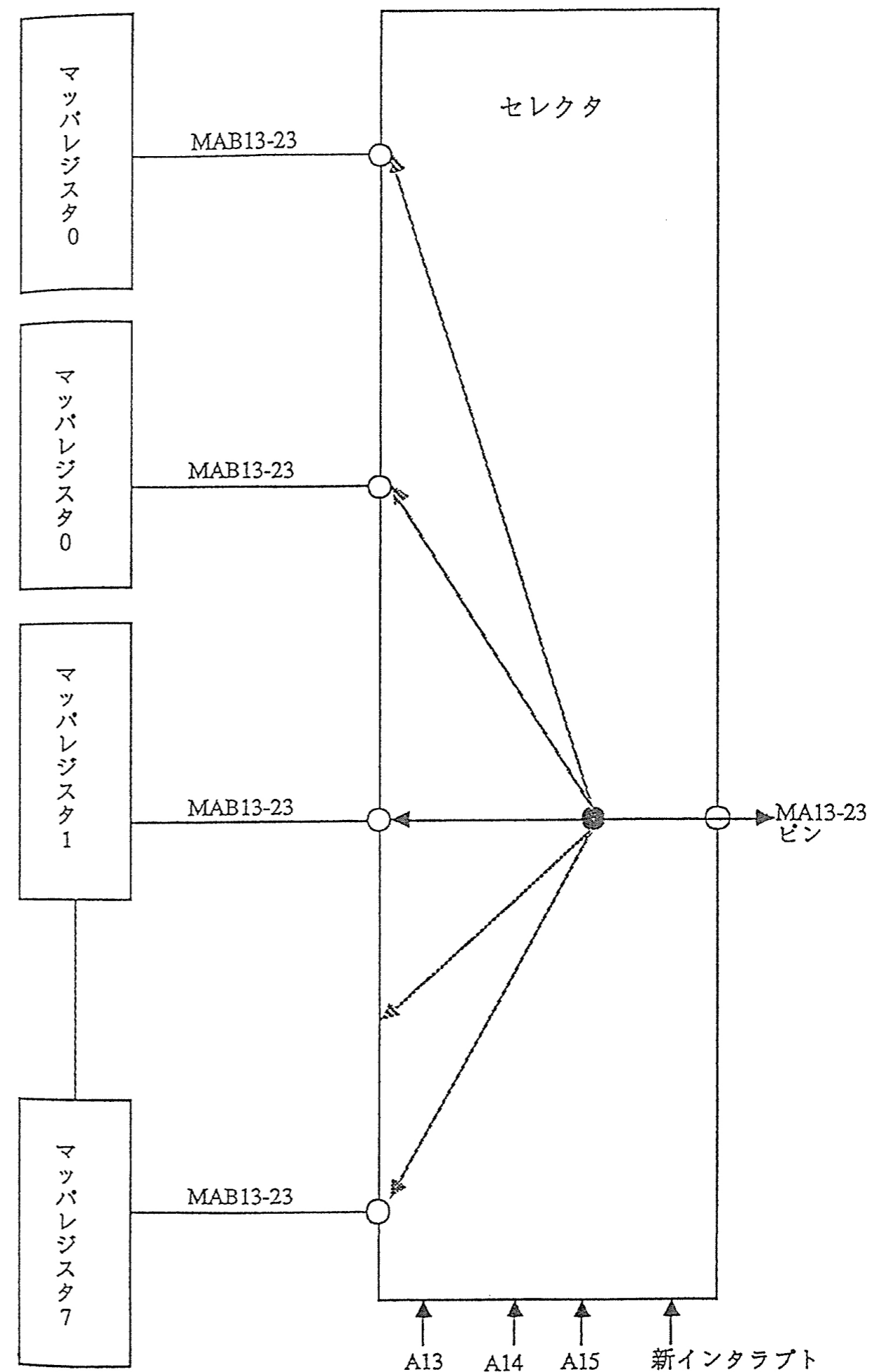
R800 では I/O データの転送命令を高速に実行できるモードがあります。このモードでは、命令のフェッチ 1 回だけで 1 バイトのデータを最高 2SYS CLK で転送します。このモードが適応される命令は、OTIR, OTDR, INIR, INDR の 4 種類です。このモードになる条件は次の通りです。

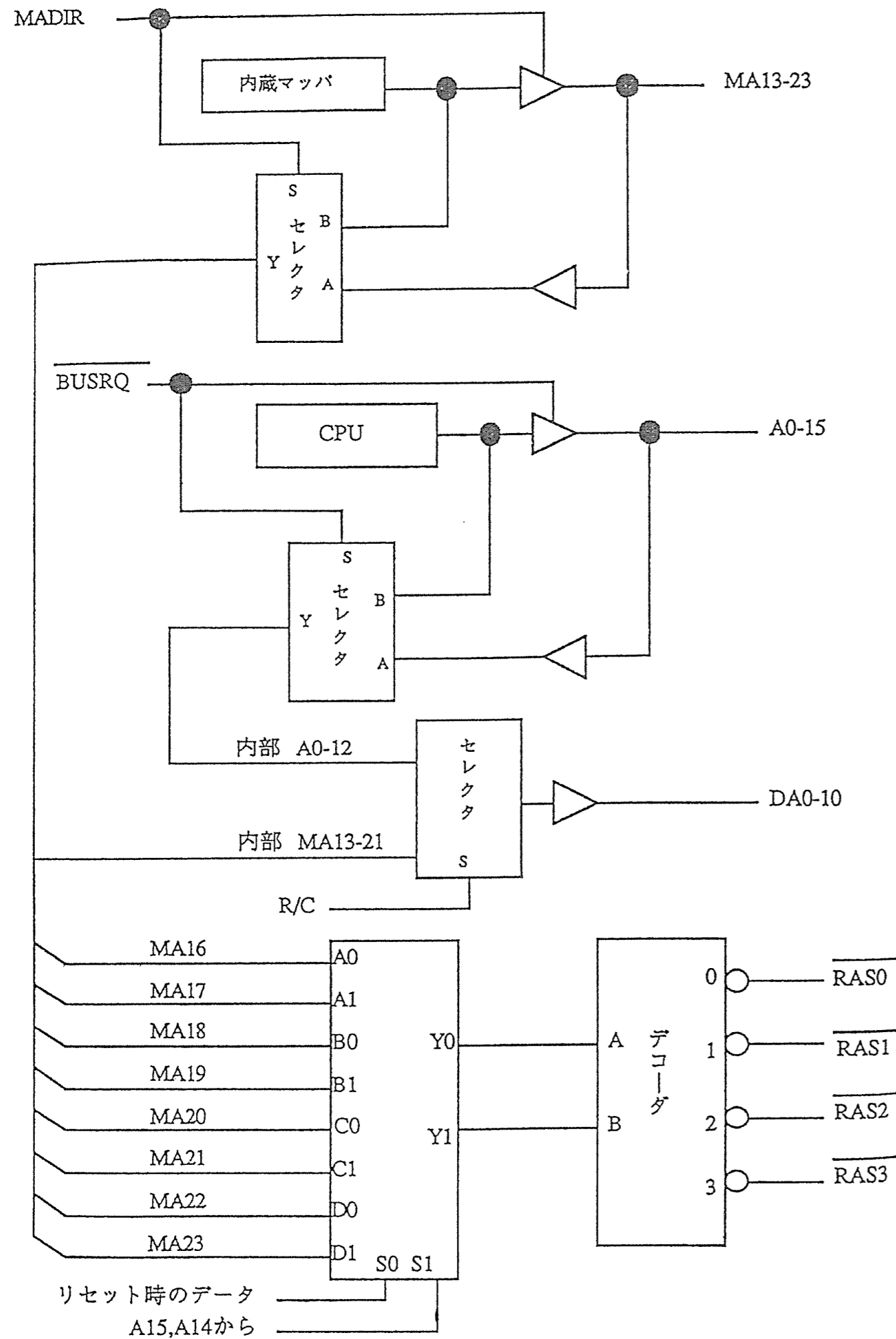
- FTREN ピンが「HIGH」であること
- インタラプトが禁止状態であること

このモードで高速 I/O アクセス中は、以下の信号は受け付けられません。

- /BUSRQ
- /INTRQ
- /NMI

R800内蔵マップ

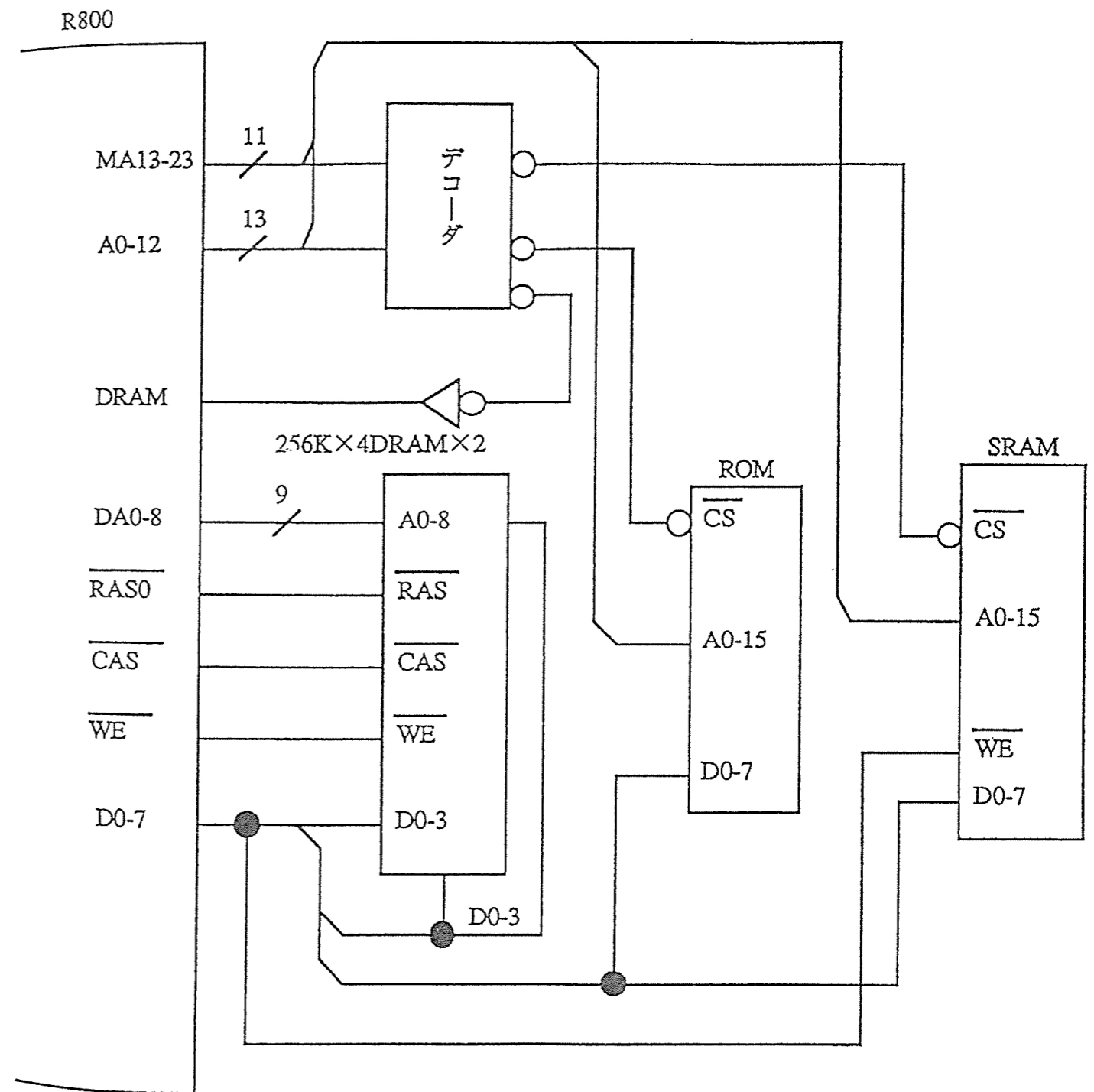




10 DRAMインターフェース

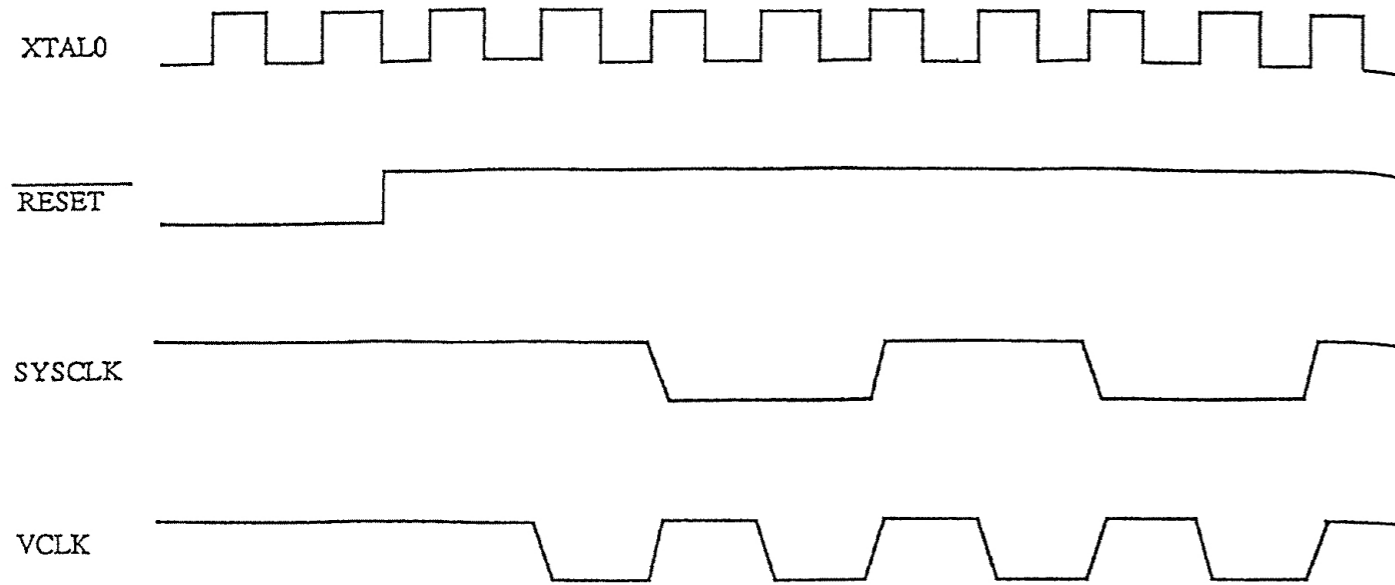
R800では命令を高速実行できるようにDRAMのページモードをサポートしており、直接接続するためのアドレスバス(DA0-DA10)を備えています。このため、メモリマップを決定する外部デコーダは直接接続したDRAMの番地をR800のDRAM信号ピンに入力する必要があります。また直接接続したDRAMはページモードで動作するため255バイト以下のマッピングは出来ません。(A0-A7はDRAM信号のデコードには用いないでください)

DRAM接続例



11 リセットとクロック

リセット中、SYSCLK、VCLK は出力されません。タイミングの詳細は図に示す通りです。



12 内部拡張レジスタ

R800 はインタラプト、DMA、マップのためのレジスタを持っています。レジスタのビットアサインなどは下記に示します。内部 I/O レジスタは CSREG ピンを「LOW」にすることでアクセスできます。アドレス信号の A0 が「LOW」の時、内部 I/O レジスタのポインターが書き込まれ「HIGH」の時データが書き込まれます。ポインターはデータ書き込み毎にオートインクリメントされるためマップレジスタ等の高速書き込みが可能です。

内部 I/O レジスタポインターレジスタ (CSREG=L, A0=L, R/W)

BIT 7
BIT 6 INTERNAL REGISTER ADDRESS 6
BIT 5 INTERNAL REGISTER ADDRESS 5
BIT 4 INTERNAL REGISTER ADDRESS 4
BIT 3 INTERNAL REGISTER ADDRESS 3
BIT 2 INTERNAL REGISTER ADDRESS 2
BIT 1 INTERNAL REGISTER ADDRESS 1
BIT 0 INTERNAL REGISTER ADDRESS 0

内部 I/O レジスタ書き込み、読み出しデータレジスタ (CSREG=L, A0=H)

BIT 7 DATA7
BIT 6 DATA6
BIT 5 DATA5
BIT 4 DATA4
BIT 3 DATA3
BIT 2 DATA2
BIT 1 DATA1
BIT 0 DATA0

***** MAPPER *****

1 MAPO (SELECTED CPU ADDRESS A13,A14,A15 = 0,0,0)

INTERNAL REGISTER ADDRESS 00H (R/W)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0

INTERNAL REGISTER ADDRESS 01H (R/W)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

2 MAP1 (SELECTED CPU ADDRESS A13,A14,A15 = 1,0,0)

INTERNAL REGISTER ADDRESS 02H (R/W)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0

INTERNAL REGISTER ADDRESS 03H (R/W)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

3 MAP2 (SELECTED CPU ADDRESS A13,A14,A15 = 0,1,0)

INTERNAL REGISTER ADDRESS 04H (R/W)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0

INTERNAL REGISTER ADDRESS 05H (R/W)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

4 MAP3 (SELECTED CPU ADDRESS A13,A14,A15 = 1,1,0)

INTERNAL REGISTER ADDRESS 06H (R/W)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0

INTERNAL REGISTER ADDRESS 07H (R/W)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

5 MAP4 (SELECTED CPU ADDRESS A13,A14,A15 = 0,0,1)

INTERNAL REGISTER ADDRESS 08H (R/W)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0

INTERNAL REGISTER ADDRESS 09H (R/W)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

6 MAP5 (SELECTED CPU ADDRESS A13,A14,A15 = 1,0,1)

INTERNAL REGISTER ADDRESS 0AH (R/W)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0

INTERNAL REGISTER ADDRESS 0BH (R/W)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

7 MAP6 (SELECTED CPU ADDRESS A13,A14,A15 = 0,1,1)

INTERNAL REGISTER ADDRESS 0CH (R/W)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0

INTERNAL REGISTER ADDRESS 0DH (R/W)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

8 MAP7 (SELECTED CPU ADDRESS A13,A14,A15 = 1,1,1)

INTERNAL REGISTER ADDRESS 0EH (R/W)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0

INTERNAL REGISTER ADDRESS 0FH (R/W)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

9 MAP0' (SELECTED CPU ADDRESS A13,A14,A15 = 0,0,0 & NEWINT SERVICE)

INTERNAL REGISTER ADDRESS 10H (R/W)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0

INTERNAL REGISTER ADDRESS 11H (R/W)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

***** DMAC *****

DMA CONTROLLER CHANNEL 0

SOURCE START ADDRESS

INTERNAL REGISTER ADDRESS 20H (WRITE ONLY)

BIT 7 A7 OUTBIT
BIT 6 A6 OUTBIT
BIT 5 A5 OUTBIT
BIT 4 A4 OUTBIT
BIT 3 A3 OUTBIT
BIT 2 A2 OUTBIT
BIT 1 A1 OUTBIT
BIT 0 A0 OUTBIT

INTERNAL REGISTER ADDRESS 21H (WRITE ONLY)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4 A12 OUTBIT
BIT 3 A11 OUTBIT
BIT 2 A10 OUTBIT
BIT 1 A9 OUTBIT
BIT 0 A8 OUTBIT

INTERNAL REGISTER ADDRESS 22H (WRITE ONLY)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

DESTINATION START ADDRESS

INTERNAL REGISTER ADDRESS 23H (WRITE ONLY)

BIT 7 A7 OUTBIT
BIT 6 A6 OUTBIT
BIT 5 A5 OUTBIT
BIT 4 A4 OUTBIT
BIT 3 A3 OUTBIT
BIT 2 A2 OUTBIT
BIT 1 A1 OUTBIT
BIT 0 A0 OUTBIT

INTERNAL REGISTER ADDRESS 24H (WRITE ONLY)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4 A12 OUTBIT
BIT 3 A11 OUTBIT
BIT 2 A10 OUTBIT
BIT 1 A9 OUTBIT
BIT 0 A8 OUTBIT

INTERNAL REGISTER ADDRESS 25H (WRITE ONLY)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

COUNTS

INTERNAL REGISTER ADDRESS 26H (WRITE ONLY)

BIT 7 C7 OUTBIT
BIT 6 C6 OUTBIT
BIT 5 C5 OUTBIT
BIT 4 C4 OUTBIT
BIT 3 C3 OUTBIT
BIT 2 C2 OUTBIT
BIT 1 C1 OUTBIT
BIT 0 C0 OUTBIT

INTERNAL REGISTER ADDRESS 27H (WRITE ONLY)

BIT 7 C15 OUTBIT
BIT 6 C14 OUTBIT
BIT 5 C13 OUTBIT
BIT 4 C12 OUTBIT
BIT 3 C11 OUTBIT
BIT 2 C10 OUTBIT
BIT 1 C9 OUTBIT
BIT 0 C8 OUTBIT

DMA MODE

INTERNAL REGISTER ADDRESS 28H

BIT 7 EOP0 (END OF PROCESS0 1:END OF PROCESS 0:BUSY) (READ ONLY)
BIT 6
BIT 5 ALD (AUTO LOAD 1:AUTO LOAD 0:NO LOAD) (WRITE ONLY)
BIT 4 EDO (ENABLE DMA0 1:ENABLE 0:DISABLE) (WRITE ONLY)
BIT 3 NCS (SOURCE ENABLE COUNT 1:COUNTUP 0:NO COUNT) (WRITE ONLY)
BIT 2 MIS (SOURCE SELECT 1:MEMORY 0:I / 0) (WRITE ONLY)
BIT 1 NCD (DESTINATION ENABLE COUNT 1:COUNTUP 0:NO COUNT) (WRITE ONLY)
BIT 0 MID (DESTINATION SELECT 1:MEMORY 0:I / 0) (WRITE ONLY)

DMA CONTROLLER CHANNEL 1

SOURCE START ADDRESS

INTERNAL REGISTER ADDRESS 30H (WRITE ONLY)

BIT 7 A7 OUTBIT
BIT 6 A6 OUTBIT
BIT 5 A5 OUTBIT
BIT 4 A4 OUTBIT
BIT 3 A3 OUTBIT
BIT 2 A2 OUTBIT
BIT 1 A1 OUTBIT
BIT 0 A0 OUTBIT

INTERNAL REGISTER ADDRESS 31H (WRITE ONLY)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4 A12 OUTBIT
BIT 3 A11 OUTBIT
BIT 2 A10 OUTBIT
BIT 1 A9 OUTBIT
BIT 0 A8 OUTBIT

INTERNAL REGISTER ADDRESS 32H (WRITE ONLY)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

DESTINATION START ADDRESS

INTERNAL REGISTER ADDRESS 33H (WRITE ONLY)

BIT 7 A7 OUTBIT
BIT 6 A6 OUTBIT
BIT 5 A5 OUTBIT
BIT 4 A4 OUTBIT
BIT 3 A3 OUTBIT
BIT 2 A2 OUTBIT
BIT 1 A1 OUTBIT
BIT 0 A0 OUTBIT

INTERNAL REGISTER ADDRESS 34H (WRITE ONLY)

BIT 7 MA15 OUTBIT
BIT 6 MA14 OUTBIT
BIT 5 MA13 OUTBIT
BIT 4 A12 OUTBIT
BIT 3 A11 OUTBIT
BIT 2 A10 OUTBIT
BIT 1 A9 OUTBIT
BIT 0 A8 OUTBIT

INTERNAL REGISTER ADDRESS 35H (WRITE ONLY)

BIT 7 MA23 OUTBIT
BIT 6 MA22 OUTBIT
BIT 5 MA21 OUTBIT
BIT 4 MA20 OUTBIT
BIT 3 MA19 OUTBIT
BIT 2 MA18 OUTBIT
BIT 1 MA17 OUTBIT
BIT 0 MA16 OUTBIT

COUNTS

INTERNAL REGISTER ADDRESS 36H (WRITE ONLY)

```

BIT 7 C7 OUTBIT
BIT 6 C6 OUTBIT
BIT 5 C5 OUTBIT
BIT 4 C4 OUTBIT
BIT 3 C3 OUTBIT
BIT 2 C2 OUTBIT
BIT 1 C1 OUTBIT
BIT 0 C0 OUTBIT

```

INTERNAL REGISTER ADDRESS 37H (WRITE ONLY)

```

BIT 7 C15 OUTBIT
BIT 6 C14 OUTBIT
BIT 5 C13 OUTBIT
BIT 4 C12 OUTBIT
BIT 3 C11 OUTBIT
BIT 2 C10 OUTBIT
BIT 1 C9 OUTBIT
BIT 0 C8 OUTBIT

```

DMA MODE

INTERNAL REGISTER ADDRESS 38H

```

BIT 7 EOP1 (END OF PROCESS1 1:END OF PROCESS 0:BUSY) (READ ONLY)
BIT 6
BIT 5 ALD (AUTO LOAD 1:AUTO LOAD 0:NO LOAD) (WRITE ONLY)
BIT 4 EDO (ENABLE DMA0 1:ENABLE 0:DISABLE) (WRITE ONLY)
BIT 3 NCS (SOURCE ENABLE COUNT 1:COUNTUP 0:NO COUNT) (WRITE ONLY)
BIT 2 MIS (SOURCE SELECT 1:MEMORY 0:I/O) (WRITE ONLY)
BIT 1 NCD (DESTINATION ENABLE COUNT 1:COUNTUP 0:NO COUNT) (WRITE ONLY)
BIT 0 MID (DESTINATION SELECT 1:MEMORY 0:I/O) (WRITE ONLY)

```

***** NEW INTERRUPT *****

INTERRUPT VECTOR

INTERNAL REGISTER ADDRESS 40H (R/W)

```

BIT 7 A7 OUTBIT
BIT 6 A6 OUTBIT
BIT 5 A5 OUTBIT
BIT 4 A4 OUTBIT
BIT 3 A3 OUTBIT
BIT 2 A2 OUTBIT
BIT 1 A1 OUTBIT
BIT 0 A0 OUTBIT

```

INTERRUPT MASK

INTERNAL REGISTER ADDRESS 41H (R/W)

```

BIT 7 NEWINT7 MASK
BIT 6 NEWINT6 MASK
BIT 5 NEWINT5 MASK
BIT 4 NEWINT4 MASK
BIT 3 NEWINT3 MASK
BIT 2 NEWINT2 MASK
BIT 1 NEWINT1 MASK
BIT 0

```

13 命令の実行

R800 は XTAL の発振周波数を 1/4 分周した SYSCLK 単位で命令を実行します。また命令のフェッチと命令の実行はパイプライン化されているので、命令の実行中に次の命令フェッチを行いません。(下記図参照) インタラプトや、バスリクエスト等により、プリフェッチした命令が実行できなくなったときは、インタラプトやバスリクエスト等のサービス終了後再び実行できなかった命令をフェッチします。

14 R800 の命令表

ここに記す命令表のクロック数は、SYSCLK 換算で XTAL の発振周波数の 4 分の 1 です。またノーウェイトで実行したときの値で、DRAM 上で実行したときはページブレイクやリフレッシュにより自動的にウェイトが挿入されます。

DRAM 上で実行するときのページブレイクは 2 つあり、1 つはアドレスの上位 8 ビットが変化したときで、もう 1 つはアドレスを決定するレジスタが変更されたときです。次に 2 つの例を示します。

1. アドレスの上位 8 ビットが変化したとき

ADDRESS	CODE
02FEH	3EH LD A,06H
02FFH	06H

ここでページブレイクが起こり 1 ウェイト挿入される

0300H	21H LD [hl],2000H
0301H	00H
0302H	20H

2. アドレスを決定するレジスタが変更されたとき

ADDRESS	CODE
020DH	21H LD [hl],2056H
020EH	56H
020FH	20H
0210H	7EH LD A,[hl]

アドレスの上位 8 ビットが変化しない実行であっても

この命令の書き込み時に 1 ウェイト挿入される

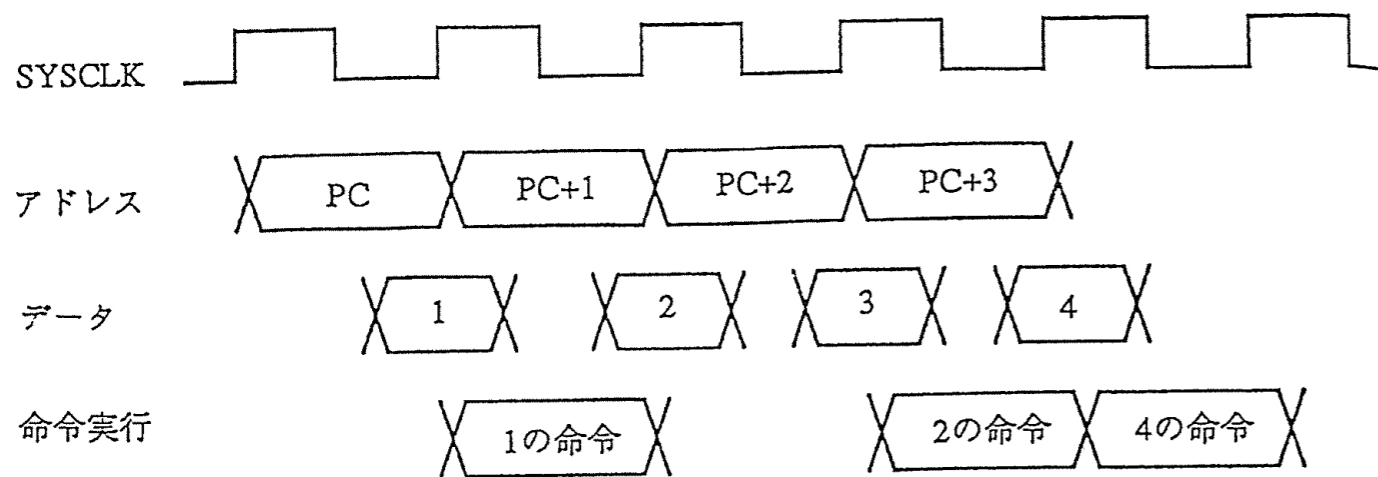
0211H	FEH CMP .a,05H
0212H	05H

注意

JP 命令 (条件 JP 含む) は実行時飛び先アドレスをテンポラリレジスタに格納するため、JP 時と飛び先命令終了後の 2 回ウェイトが挿入されます。すなわち DRAM 上で JP 命令を実行したとき 5 クロックかかることとなります (条件ジャンプでジャンプしないときは 3 クロックです)。また JR 命令ではテンポラリレジスタが用いられていないため、このウェイトは挿入されません。

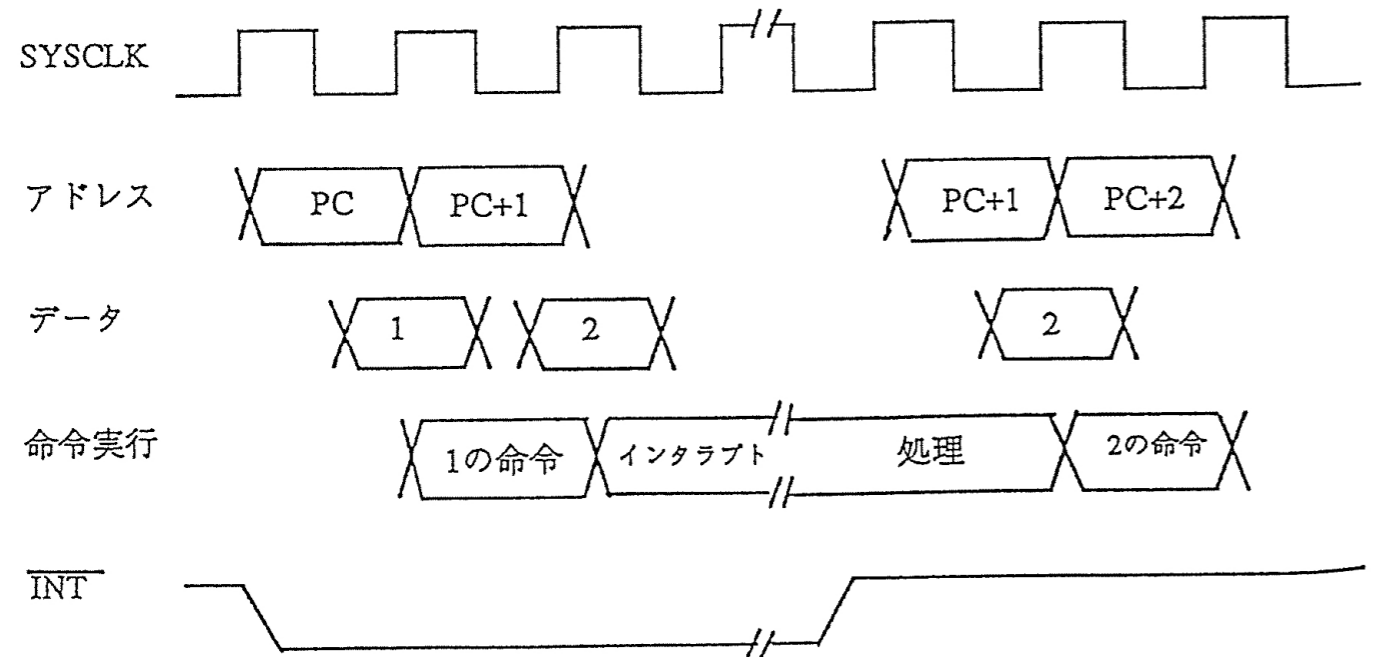
リフレッシュのサイクル数

リフレッシュそのもののサイクル数は 2 クロックですが、リフレッシュの前にプリフェッチした命令を再度フェッチするためさらに 2 クロックがかかり、合計 4 クロックになります。



※ 1、2の命令は1バイト命令

※ 2の命令は2バイト命令



凡例

- .a(r) レジスタ.aの最上位ビット
- .a{4..7} レジスタ.aのビット4-7
- ; 動作の区切り
- .de:hl 上位16ビットが.de 下位16ビットが.hl にはいる32ビット整数
- [.ix+d] .ixに8ビットの符号つき変位を足した値が示すアドレス
- C キャリーフラグ
- Z ゼロフラグ
- P_v パリティ・オーバーフローフラグ
- S サインフラグ
- N 減算フラグ
- H ハーフキャリーフラグ
- フラグは変化無し
- ! フラグは結果が反映されます
- 0 フラグは0
- 1 フラグは1
- ? 不定になります
- V オーバーフローフラグとして使われます
- P パリティフラグとして使われます
- ≡ 割り込みフリップフロップの値が入ります
- r,r' 8ビットレジスタ、.a,.b,.c,.d,.e,.h,.l
- u,u' 8ビットレジスタ、.a,.b,.c,.d,.e,.ixh,.ixl
- v,v' 8ビットレジスタ、.a,.b,.c,.d,.e,.iyh,.iyl
- p 8ビットレジスタ、.ixh,.ixl
- q 8ビットレジスタ、.iyh,.iyl
- ss 16ビットレジスタ、.bc,.de,.hl,.sp
- pp 16ビットレジスタ、.bc,.de,.ix,.sp
- rr 16ビットレジスタ、.bc,.de,.iy,.sp
- qq 16ビットレジスタ、.bc,.de,.hl,.af
- e short br系の命令の飛び先アドレスへの差分、8ビットの符号つき即値 (+127~-128)
- k brk命令の飛び先アドレス、00h,08h,10h,18h,20h,28h,30h,38h
- nn 16ビットの即値、もしくは絶対アドレス
- n 8ビットの即値
- b ビット演算命令の第何ビットかを示す値
- NOT ビットを反転します
- V ビットのORをとります
- ∨ ビットのXORをとります
- ∧ ビットのANDをとります
- tmp 一時的に値を待避します

分岐命令、コール命令でクロック数が2つかいてあるものは
 上が条件が成立しない時、下が条件が成立した時です
 入出力命令でクロック数が2つかいてあるものは
 上がまだ転送が終わらない時、下が転送が終わった時です

8ビット移動命令						
ニーモニック	命令動作	flags	オペコード		bytes	clocks
			S Z H P ₂ N C	Hex		
ld r,r'	r←r'	01 r r'		1	1
ld r,n	r←n	00 r 110 ← n →		2	2
ld r,[hl]	r←[hl]	01 r 110		1	2
ld r,[ix+d]	r←[ix+d]	11011101 01 r 110 ← d →	DD	3	5
ld r,[iy+d]	r←[iy+d]	11111101 01 r 110 ← d →	FD	3	5
ld [hl],r	[hl]←r	01110 r		1	2
ld [ix+d],r	[ix+d]←r	11011101 01110 r ← d →	DD	3	5
ld [iy+d],r	[iy+d]←r	11111101 01110 r ← d →	FD	3	5
ld u,u'	u←u'	11011101 01 u u'	DD	2	2
ld v,v'	v←v'	11111101 01 v v'	FD	2	2
ld u,n	u←n	11011101 00 u 110 ← n →	DD	3	3
ld v,n	v←n	11111101 00 v 110 ← n →	FD	3	3
ld [hl],n	[hl]←n	00110110 ← n →	36	2	3
ld [ix+d],n	[ix+d]←n	11011101 00110110 ← d → ← n →	DD 36	4	5
ld [iy+d],n	[iy+d]←n	11111101 00110110 ← d → ← n →	FD 36	4	5
ld .a,i	.a←i	↑ ↓ 0 0 0 0	11101101 01010111	ED 57	2	2
ld .a,r	.a←r	↑ ↓ 0 0 0 0	11101101 01011111	ED 5F	2	2
ld .i,a	.i←a	11101101 01000111	ED 47	2	2
ld .r,a	.r←a	11101101 01001111	ED 4F	2	2
ld .a,[bc]	.a←[bc]	00001010	0A	1	2
ld .a,[de]	.a←[de]	00011010	1A	1	2
ld .a,[nn]	.a←[nn]	00111010 ← nn ₁ → ← nn ₂ →	3A	3	4
ld [bc],.a	[bc]←.a	000001010	02	1	2
ld [de],.a	[de]←.a	000101010	12	1	2
ld [nn],.a	[nn]←.a	001101010 ← nn ₁ → ← nn ₂ →	32	3	4

000	001	010	011	100	101	110	111
r .b	.c	.d	.e	.h	.l		.a
u .b	.c	.d	.e	.ixh	.ixl		.a
v .b	.c	.d	.e	.iyh	.iyl		.a

16ビット移動命令						
ニーモニック	命令動作	flags S Z H P _v N C	オペコード		bytes	clocks
			76543210	Hex		
ld ss,nn	ss←nn	00..0001		3	3
			←nn _L →			
			←nn _H →			
ld ix,nn	ix←nn	11011101	DD	4	4
			00100001	21		
			←nn _L →			
			←nn _H →			
ld iy,nn	iy←nn	11111101	FD	4	4
			00100001	21		
			←nn _L →			
			←nn _H →			
ld sp,.hl	sp←.hl	11111001	F9	1	1
ld sp,ix	sp←ix	11011101	DD	2	2
			11111001	F9		
ld sp,iy	sp←iy	11111101	FD	2	2
			11111001	F9		
ld ss,[nn]	ss _H ←[nn+1] ss _L ←[nn]	11101101	ED	4	6
			01..1011			
			←nn _L →			
			←nn _H →			
ld hl,[nn]	.h←[nn+1] .l←[nn]	00101010	2A	3	5
			←nn _L →			
			←nn _H →			
ld ix,[nn]	.ixh←[nn+1] .ixl←[nn]	11011101	DD	4	6
			00101010	2A		
			←nn _L →			
			←nn _H →			
ld iy,[nn]	.iyh←[nn+1] .iyl←[nn]	11111101	FD	4	6
			00101010	2A		
			←nn _L →			
			←nn _H →			
ld [nn],ss	[nn+1]←ss _H [nn]←ss _L	11101101	ED	4	6
			01..0011			
			←nn _L →			
			←nn _H →			
ld [nn],.hl	[nn+1]←.h [nn]←.l	00100010	22	3	5
			←nn _L →			
			←nn _H →			
ld [nn],.ix	[nn+1]←.ixh [nn]←.ixl	11011101	DD	4	6
			00100010	22		
			←nn _L →			
			←nn _H →			
ld [nn],.iy	[nn+1]←.iyh [nn]←.iyl	11111101	FD	4	6
			00100010	22		
			←nn _L →			
			←nn _H →			

00 01 10 11
ss.bc.de.hl.sp

交換命令						
ニーモニック	命令動作	flags S Z H P _v N C	オペコード		bytes	clocks
			76543210	Hex		
xch .de,.hl	.de↔.hl	11101011	EB	1	1
xch .af,.af'	.af↔.af'	↑↑↑↑↑↑	00001000	08	1	1
xch [sp],.hl	.l↔[sp];.h↔[sp+1]	11100011	E3	1	5
xch [sp],.ix	.ixl↔[sp] .ixh↔[sp+1]	11011101	DD	2	6
			11100011	E3		
xch [sp],.iy	.iyl↔[sp] .iyh↔[sp+1]	11111101	FD	2	6
			11100011	E3		
xchx	.bc↔.bc';.de↔.de';.hl↔.hl'	11011001	D9	1	1

スタック操作命令						
ニーモニック	命令動作	flags S Z H P _v N C	オペコード		bytes	clocks
			76543210	Hex		
push qq	[sp-2]←qq _L ;[sp-1]←qq _H sp←sp-2	11 qq 0101		1	4
push ix	[sp-2]←.ixl;[sp-1]←.ixh sp←sp-2	11011101	DD	2	5
			11100101	E5		
push iy	[sp-2]←.iyl;[sp-1]←.iyh sp←sp-2	11111101	FD	2	5
			11100101	E5		
pop qq	qq _L ←[sp];qq _H ←[sp+1] sp←sp+2	11 qq 0001		1	3
pop ix	.ixl←[sp];.ixh←[sp+1] sp←sp+2	11011101	DD	2	4
			11100001	E1		
pop iy	.iyl←[sp];.iyh←[sp+1] sp←sp+2	11111101	FD	2	4
			11100001	E1		

00 01 10 11
qq.bc.de.hl.af pop.afの時はflagsはすべて変化します

ブロック転送命令						
ニーモニック	命令動作	flags S Z H P _v N C	オペコード		bytes	clocks
			76543210	Hex		
move [hl++],[de++]	[de]←[hl];.de←.de+1 .hl←.hl+1;.bc←.bc-1	..0100	11101101	ED	2	4
		.1	10100000	A0		
move [hl--],[de--]	[de]←[hl];.de←.de-1 .hl←.hl-1;.bc←.bc-1	..0100	11101101	ED	2	4
		.1	10101000	A8		
movem [hl++],[de++]	repeat;[de]←[hl];.de←.de+1 .hl←.hl+1;.bc←.bc-1;until .bc=0	..0000	11101101	ED	2	4
			10110000	B0		
movem [hl--],[de--]	repeat;[de]←[hl];.de←.de-1 .hl←.hl-1;.bc←.bc-1;until .bc=0	..0000	11101101	ED	2	4
			10111000	B8		

.1.bc-1=0の時0 その他1

ブロックサーチ命令						
ニーモニック	命令動作	flags S Z H P _v N C	オペコード		bytes	clocks
			76543210	Hex		
cmp .a,[hl++]	.a-[hl];.hl←.hl+1 .bc←.bc-1	↑↑↑↑10	11101101	ED	2	4
		.2 .1	10100001	A1		
cmp .a,[hl--]	.a-[hl];.hl←.hl-1 .bc←.bc-1	↑↑↑↑10	11101101	ED	2	4
		.2 .1	10101001	A9		
cmpm .a,[hl++]	repeat;.a-[hl];.hl←.hl+1 .bc←.bc-1;until .bc=0 or .a=[hl]	↑↑↑↑10	11101101	ED	2	5
		.2 .1	10110001	B1		
cmpm .a,[hl--]	repeat;.a-[hl];.hl←.hl-1 .bc←.bc-1;until .bc=0 or .a=[hl]	↑↑↑↑10	11101101	ED	2	5
		.2 .1	10111001	B9		

.1.bc-1=0の時0 その他1

.2.a=[hl]の時1 その他0

加算命令						
ニーモニック	命令動作	flags S Z H ^o N C	オペコード		bytes	clocks
			78543210	Hex		
add .a,r	.a←.a+r	↑↑↑V0↑	10000	r	1	1
add .a,p	.a←.a+p	↑↑↑V0↑	11011101	DD	2	2
			10000	p		
add .a,q	.a←.a+q	↑↑↑V0↑	11111101	FD	2	2
			10000	q		
add .a,[.hl]	.a←.a+[.hl]	↑↑↑V0↑	10000110	86	1	2
add .a,[.ix+d]	.a←.a+[.ix+d]	↑↑↑V0↑	11011101	DD	3	5
			10000110	86		
			← d →			
add .a,[.iy+d]	.a←.a+[.iy+d]	↑↑↑V0↑	11111101	FD	3	5
			10000110	86		
			← d →			
add .a,n	.a←.a+n	↑↑↑V0↑	11000110	C6	2	2
			← n →			
addc .a,r	.a←.a+r+c	↑↑↑V0↑	10001	r	1	1
addc .a,p	.a←.a+p+c	↑↑↑V0↑	11011101	DD	2	2
			10001	p		
addc .a,q	.a←.a+q+c	↑↑↑V0↑	11111101	FD	2	2
			10001	q		
addc .a,[.hl]	.a←.a+[.hl]+c	↑↑↑V0↑	10001110	8E	1	2
addc .a,[.ix+d]	.a←.a+[.ix+d]+c	↑↑↑V0↑	11011101	DD	3	5
			10001110	8E		
			← d →			
addc .a,[.iy+d]	.a←.a+[.iy+d]+c	↑↑↑V0↑	11111101	FD	3	5
			10001110	8E		
			← d →			
addc .a,n	.a←.a+n+c	↑↑↑V0↑	11001110	CE	2	2
			← n →			
addc .hl,ss	.hl←.hl+ss+c	↑↑?V0↑	11101101	ED	2	2
			01..1010			
add .hl,ss	.hl←.hl+ss	...?0↑	00..1001		1	1
add .ix,pp	.ix←.ix+pp	...?0↑	11011101	DD	2	2
			00pp1001			
add .iy,rr	.iy←.iy+rr	...?0↑	11111101	FD	2	2
			00rr1001			
inc r	r←r+1	↑↑↑V0•	00	r 100	1	1
inc p	p←p+1	↑↑↑V0•	11011101	DD	2	2
			00	p 100		
inc q	q←q+1	↑↑↑V0•	11111101	FD	2	2
			00	q 100		
inc [.hl]	[.hl]←[.hl]+1	↑↑↑V0•	00110100	34	1	4
inc [.ix+d]	[.ix+d]←[.ix+d]+1	↑↑↑V0•	11011101	DD	3	7
			00110100	34		
			← d →			
inc [.iy+d]	[.iy+d]←[.iy+d]+1	↑↑↑V0•	11111101	FD	3	7
			00110100	34		
			← d →			
inc ss	ss←ss+1	00..0011		1	1
inc .ix	.ix←.ix+1	11011101	DD	2	2
			00100011	23		
inc .iy	.iy←.iy+1	11111101	FD	2	2
			00100011	23		

00	01	10	11	
ss	.bc	.de	.hl	.sp
pp	.bc	.de	.ix	.sp
rr	.bc	.de	.iy	.sp

000001	010011	100101	110111
p		.ixh	.ixl
q		.iyh	.iyl

減算命令						
ニーモニック	命令動作	flags S Z H P ₀ N C	オペコード			
			76543210	Hex	bytes	clocks
sub .a,r	.a←.a-r	↑↑↑V1↑	10010 r		1	1
sub .a,p	.a←.a-p	↑↑↑V1↑	11011101 10010 p	DD	2	2
sub .a,q	.a←.a-q	↑↑↑V1↑	11111101 10010 q	FD	2	2
sub .a,[.hl]	.a←.a-[.hl]	↑↑↑V1↑	10010110	96	1	2
sub .a,[ix+d]	.a←.a-[ix+d]	↑↑↑V1↑	11011101 10010110 ← d →	DD 96	3	5
sub .a,[iy+d]	.a←.a-[iy+d]	↑↑↑V1↑	11111101 10010110 ← d →	FD 96	3	5
sub .a,n	.a←.a-n	↑↑↑V1↑	11010110 ← n →	D6	2	2
subc.a,r	.a←.a-r-c	↑↑↑V1↑	10011 r		1	1
subc.a,p	.a←.a-p-c	↑↑↑V1↑	11011101 10011 p	DD	2	2
subc.a,q	.a←.a-q-c	↑↑↑V1↑	11111101 10011 q	FD	2	2
subc.a,[.hl]	.a←.a-[.hl]-c	↑↑↑V1↑	10011110	9E	1	2
subc.a,[ix+d]	.a←.a-[ix+d]-c	↑↑↑V1↑	11011101 10011110 ← d →	DD 9E	3	5
subc.a,[iy+d]	.a←.a-[iy+d]-c	↑↑↑V1↑	11111101 10011110 ← d →	FD 9E	3	5
subc.a,n	.a←.a-n-c	↑↑↑V1↑	11011110 ← n →	DE	2	2
subc.hl,ss	.hl←.hl-ss-c	↑↑?V1↑	11101101 01..0010	ED	2	2
dec r	r←r-1	↑↑↑V1●	00 r 101		1	1
dec p	p←p-1	↑↑↑V1●	11011101 00 p 101	DD	2	2
dec q	q←q-1	↑↑↑V1●	11111101 00 q 101	FD	2	2
dec [.hl]	[.hl]←[.hl]-1	↑↑↑V1●	00110101	35	1	4
dec [ix+d]	[ix+d]←[ix+d]-1	↑↑↑V1●	11011101 00110101 ← d →	DD 35	3	7
dec [iy+d]	[iy+d]←[iy+d]-1	↑↑↑V1●	11111101 00110101 ← d →	FD 35	3	7
dec ss	ss←ss-1	●●●●●●	00..1011		1	1
dec ix	ix←ix-1	●●●●●●	11011101 00101011	DD 2B	2	2
dec iy	iy←iy-1	●●●●●●	11111101 00101011	FD 2B	2	2

論理演算命令						
ニーモニック	命令動作	flags S Z H P ₀ N C	オペコード			
			76543210	Hex	bytes	clocks
and .a,r	.a←.a^r	↑↑↑P00	10100 r		1	1
and .a,p	.a←.a^p	↑↑↑P00	11011101 10100 p	DD	2	2
and .a,q	.a←.a^q	↑↑↑P00	11111101 10100 q	FD	2	2
and .a,[.hl]	.a←.a^[.hl]	↑↑↑P00	10100110	A6	1	2
and .a,[ix+d]	.a←.a^[ix+d]	↑↑↑P00	11011101 10100110 ← d →	DD A6	3	5
and .a,[iy+d]	.a←.a^[iy+d]	↑↑↑P00	11111101 10100110 ← d →	FD A6	3	5
and .a,n	.a←.a^n	↑↑↑P00	11100110 ← n →	E6	2	2
or .a,r	.a←.a^r	↑↑0P00	10110 r		1	1
or .a,p	.a←.a^p	↑↑0P00	11011101 10110 p	DD	2	2
or .a,q	.a←.a^q	↑↑0P00	11111101 10110 q	FD	2	2
or .a,[.hl]	.a←.a^[.hl]	↑↑0P00	10110110	B6	1	2
or .a,[ix+d]	.a←.a^[ix+d]	↑↑0P00	11011101 10110110 ← d →	DD B6	3	5
or .a,[iy+d]	.a←.a^[iy+d]	↑↑0P00	11111101 10110110 ← d →	FD B6	3	5
or .a,n	.a←.a^n	↑↑0P00	11110110 ← n →	F6	2	2
xor .a,r	.a←.a^r	↑↑0P00	10101 r		1	1
xor .a,p	.a←.a^p	↑↑0P00	11011101 10101 p	DD	2	2
xor .a,q	.a←.a^q	↑↑0P00	11111101 10101 q	FD	2	2
xor .a,[.hl]	.a←.a^[.hl]	↑↑0P00	10101110	AE	1	2
xor .a,[ix+d]	.a←.a^[ix+d]	↑↑0P00	11011101 10101110 ← d →	DD AE	3	5
xor .a,[iy+d]	.a←.a^[iy+d]	↑↑0P00	11111101 10101110 ← d →	FD AE	3	5
xor .a,n	.a←.a^n	↑↑0P00	11101110 ← n →	EE	2	2
cmp .a,r	.a-r	↑↑↑V1↑	10111 r		1	1
cmp .a,p	.a-p	↑↑↑V1↑	11011101 10111 p	DD	2	2
cmp .a,q	.a-q	↑↑↑V1↑	11111101 10111 q	FD	2	2
cmp .a,[.hl]	.a-[.hl]	↑↑↑V1↑	10111110	BE	1	2
cmp .a,[ix+d]	.a-[ix+d]	↑↑↑V1↑	11011101 10111110 ← d →	DD BE	3	5
cmp .a,[iy+d]	.a-[iy+d]	↑↑↑V1↑	11111101 10111110 ← d →	FD BE	3	5
cmp .a,n	.a-n	↑↑↑V1↑	11111110 ← n →	FE	2	2

乗算命令						
ニーモニク	命令動作	flags	オペコード	bytes	clocks	
		S Z H P _N C	76 543 210 Hex			
mulub .a,r	.hl←.a*r	0 1 0 0 1	11 101 101 11 r 001	ED	2	14
muluw .hl,ss	.de:.hl←.hl*ss	0 1 0 0 1	11 101 101 11 .a 0011	ED	2	36

mulub では r が b,c,d,e の時以外は動作が保証されません
muluw では ss が bc,sp の時以外は動作が保証されません

ビット操作命令						
ニーモニク	命令動作	flags	オペコード	bytes	clocks	
		S Z H P _N C	76 543 210 Hex			
bit b,r	z←NOT r _(b)	? 1 1 ? 0 0	11 001 011 01 b r	CB	2	2
bit b,[hl]	z←NOT [hl] _(b)	? 1 1 ? 0 0	11 001 011 01 b 110	CB	2	3
bit b,[ix+d]	z←NOT [ix+d] _(b)	? 1 1 ? 0 0	11 011 101 11 001 011 ← d → 01 b 110	DD CB	4	5
bit b,[iy+d]	z←NOT [iy+d] _(b)	? 1 1 ? 0 0	11 111 101 11 001 011 ← d → 01 b 110	FD CB	4	5
set b,r	r _(b) ←1	0 0 0 0 0	11 001 011 11 b r	CB	2	2
set b,[hl]	[hl] _(b) ←1	0 0 0 0 0	11 001 011 11 b 110	CB	2	5
set b,[ix+d]	[ix+d] _(b) ←1	0 0 0 0 0	11 011 101 11 001 011 ← d → 11 b 110	DD CB	4	7
set b,[iy+d]	[iy+d] _(b) ←1	0 0 0 0 0	11 111 101 11 001 011 ← d → 11 b 110	FD CB	4	7
clr b,r	r _(b) ←0	0 0 0 0 0	11 001 011 10 b r	CB	2	2
clr b,[hl]	[hl] _(b) ←0	0 0 0 0 0	11 001 011 10 b 110	CB	2	5
clr b,[ix+d]	[ix+d] _(b) ←0	0 0 0 0 0	11 011 101 11 001 011 ← d → 10 b 110	DD CB	4	7
clr b,[iy+d]	[iy+d] _(b) ←0	0 0 0 0 0	11 111 101 11 001 011 ← d → 10 b 110	FD CB	4	7

ローテイト命令						
ニーモニク	命令動作	flags	オペコード	bytes	clocks	
		S Z H P _N C	76 543 210 Hex			
rola	C←.a _(r) ;a←.a*2;a ₍₀₎ ←C	0 0 0 0 1	00 000 111	07	1	1
rora	C←.a ₍₀₎ ;a←.a/2;a _(r) ←C	0 0 0 0 1	00 000 111	0F	1	1
rolca	tmp←C;C←.a _(r) ;a←.a*2;a ₍₀₎ ←tmp	0 0 0 0 1	00 010 111	17	1	1
rorca	tmp←C;C←.a ₍₀₎ ;a←.a/2;a _(r) ←tmp	0 0 0 0 1	00 011 111	1F	1	1
rol r	C←r _(r) r←r*2;r ₍₀₎ ←C	1 1 0 P 0 1	11 001 011 00 000 r	CB	2	2
rol [hl]	C←[hl] _(r) [hl]←[hl]*2;[hl] ₍₀₎ ←C	1 1 0 P 0 1	11 001 011 00 000 110	CB	2	5
rol [ix+d]	C←[ix+d] _(r) [ix+d]←[ix+d]*2 [ix+d] ₍₀₎ ←C	1 1 0 P 0 1	11 011 101 11 001 011 ← d → 00 000 110	DD CB	4	7
rol [iy+d]	C←[iy+d] _(r) [iy+d]←[iy+d]*2 [iy+d] ₍₀₎ ←C	1 1 0 P 0 1	11 111 101 11 001 011 ← d → 00 000 110	FD CB	4	7
ror r	C←r ₍₀₎ r←r/2;r _(r) ←C	1 1 0 P 0 1	11 001 011 00 001 r	CB	2	2
ror [hl]	C←[hl] ₍₀₎ [hl]←[hl]/2;[hl] _(r) ←C	1 1 0 P 0 1	11 001 011 00 001 110	CB	2	5
ror [ix+d]	C←[ix+d] ₍₀₎ [ix+d]←[ix+d]/2 [ix+d] _(r) ←C	1 1 0 P 0 1	11 011 101 11 001 011 ← d → 00 001 110	DD CB	4	7
ror [iy+d]	C←[iy+d] ₍₀₎ [iy+d]←[iy+d]/2 [iy+d] _(r) ←C	1 1 0 P 0 1	11 111 101 11 001 011 ← d → 00 001 110	FD CB	4	7
rolc r	tmp←C;C←r _(r) r←r*2;r ₍₀₎ ←tmp	1 1 0 P 0 1	11 001 011 00 010 r	CB	2	2
rolc [hl]	tmp←C;C←[hl] _(r) [hl]←[hl]*2;[hl] ₍₀₎ ←tmp	1 1 0 P 0 1	11 001 011 00 010 110	CB	2	5
rolc [ix+d]	tmp←C C←[ix+d] _(r) [ix+d]←[ix+d]*2 [ix+d] ₍₀₎ ←tmp	1 1 0 P 0 1	11 011 101 11 001 011 ← d → 00 010 110	DD CB	4	7
rolc [iy+d]	tmp←C C←[iy+d] _(r) [iy+d]←[iy+d]*2 [iy+d] ₍₀₎ ←tmp	1 1 0 P 0 1	11 111 101 11 001 011 ← d → 00 010 110	FD CB	4	7
rorc r	tmp←C;C←r ₍₀₎ r←r/2;r _(r) ←tmp	1 1 0 P 0 1	11 001 011 00 011 r	CB	2	2
rorc [hl]	tmp←C;C←[hl] ₍₀₎ [hl]←[hl]/2;[hl] _(r) ←tmp	1 1 0 P 0 1	11 001 011 00 011 110	CB	2	5
rorc [ix+d]	tmp←C C←[ix+d] ₍₀₎ [ix+d]←[ix+d]/2 [ix+d] _(r) ←tmp	1 1 0 P 0 1	11 011 101 11 001 011 ← d → 00 011 110	DD CB	4	7
rorc [iy+d]	tmp←C C←[iy+d] ₍₀₎ [iy+d]←[iy+d]/2 [iy+d] _(r) ←tmp	1 1 0 P 0 1	11 111 101 11 001 011 ← d → 00 011 110	FD CB	4	7
rol4 [hl]	tmp←.a _(0..3) ;a _(0..3) ←[hl] _(4..7) [hl] _(4..7) ←[hl] _(0..3) ;[hl] _(0..3) ←tmp	1 1 0 P 0 0	11 101 101 11 101 111	ED 6F	2	5
ror4 [hl]	tmp←.a _(0..3) ;a _(0..3) ←[hl] _(0..3) [hl] _(0..3) ←[hl] _(4..7) ;[hl] _(4..7) ←tmp	1 1 0 P 0 0	11 101 101 11 100 111	ED 67	2	5

シフト命令						
ニーモニック	命令動作	flags S Z H P _v N C	オペコード		bytes	clocks
			76543210	Hex		
shl r shla	C←r{7} r←r*2	↑↑0P0↑	11001011 00100 r	CB	2	2
shl [.hl] shla	C←[.hl]{7} [.hl]←[.hl]*2	↑↑0P0↑	11001011 00100110	CB 26	2	5
shl [.ix+d] shla	C←[.ix+d]{7} [.ix+d]←[.ix+d]*2	↑↑0P0↑	11011101 11001011 ← d → 00100110	DD CB 26	4	7
shl [.iy+d] shla	C←[.iy+d]{7} [.iy+d]←[.iy+d]*2	↑↑0P0↑	11111101 11001011 ← d → 00100110	FD CB 26	4	7
shr r	C←r{0} r←r/2	↑↑0P0↑	11001011 00111 r	CB	2	2
shr [.hl]	C←[.hl]{0} [.hl]←[.hl]/2	↑↑0P0↑	11001011 00111110	CB 3E	2	5
shr [.ix+d]	C←[.ix+d]{0} [.ix+d]←[.ix+d]/2	↑↑0P0↑	11011101 11001011 ← d → 00111110	DD CB 3E	4	7
shr [.iy+d]	C←[.iy+d]{0} [.iy+d]←[.iy+d]/2	↑↑0P0↑	11111101 11001011 ← d → 00111110	FD CB 3E	4	7
shrar	tmp←r{7};C←r{0} r←r/2;r{7}←tmp	↑↑0P0↑	11001011 00101 r	CB	2	2
shra [.hl]	tmp←[.hl]{7};C←[.hl]{0} [.hl]←[.hl]/2;[.hl]{7}←tmp	↑↑0P0↑	11001011 00101110	CB 2E	2	5
shra [.ix+d]	tmp←[.ix+d]{7} C←[.ix+d]{0} [.ix+d]←[.ix+d]/2 [.ix+d]{7}←tmp	↑↑0P0↑	11011101 11001011 ← d → 00101110	DD CB 2E	4	7
shra [.iy+d]	tmp←[.iy+d]{7} C←[.iy+d]{0} [.iy+d]←[.iy+d]/2 [.iy+d]{7}←tmp	↑↑0P0↑	11111101 11001011 ← d → 00101110	FD CB 2E	4	7

shl 命令と shla 命令はまったく同じ物なのでオペランドは同一です

分岐命令						
ニーモニック	命令動作	flags S Z H P _v N C	オペコード		bytes	clocks
			76543210	Hex		
br nn	.pc←nn	11000011 ← nn ₁ → ← nn ₈ →	C3	3	3
bnz nn	if z=0 .pc←nn	11000010 ← nn ₁ → ← nn ₈ →	C2	3	3
bz nn	if z=1 .pc←nn	11001010 ← nn ₁ → ← nn ₈ →	CA	3	3
bnc nn	if c=0 .pc←nn	11010010 ← nn ₁ → ← nn ₈ →	D2	3	3
bc nn	if c=1 .pc←nn	11011010 ← nn ₁ → ← nn ₈ →	DA	3	3
bpo nn	if P _v =0 .pc←nn	11100010 ← nn ₁ → ← nn ₈ →	E2	3	3
bpe nn	if P _v =1 .pc←nn	11101010 ← nn ₁ → ← nn ₈ →	EA	3	3
bp nn	if s=0 .pc←nn	11110010 ← nn ₁ → ← nn ₈ →	F2	3	3
bm nn	if s=1 .pc←nn	11111010 ← nn ₁ → ← nn ₈ →	FA	3	3
br [.hl]	.pc←[.hl]	11101001	E9	1	1
br [.ix]	.pc←[.ix]	11011101 11101001	DD E9	2	2
br [.iy]	.pc←[.iy]	11111101 11101001	FD E9	2	2
short br e	.pc←.pc+e	00011000 ← e-2 →	18	2	3
short bnz e	if z=0 .pc←.pc+e	00100000 ← e-2 →	20	2	2 3
short bz e	if z=1 .pc←.pc+e	00101000 ← e-2 →	28	2	2 3
short bnc e	if c=0 .pc←.pc+e	00110000 ← e-2 →	30	2	2 3
short bc e	if c=1 .pc←.pc+e	00111000 ← e-2 →	38	2	2 3
dbnze	.b←.b-1;if .b≠0 .pc←.pc+e	00010000 ← e-2 →	10	2	2

コール命令						
ニーモニック	命令動作	flags S Z H P _N C	オペコード		bytes	clocks
			76543210	Hex		
call nn	[.sp-2]←.pc _L ; [.sp-1]←.pc _H .sp←.sp-2; pc←nn	••••••	11001101	CD	3	5
call nz,nn	if z=0 [.sp-2]←.pc _L ; [.sp-1]←.pc _H .sp←.sp-2; pc←nn	••••••	11000100	C4	3	3
call z,nn	if z=1 [.sp-2]←.pc _L ; [.sp-1]←.pc _H .sp←.sp-2; pc←nn	••••••	11001100	CC	3	3
call nc,nn	if c=0 [.sp-2]←.pc _L ; [.sp-1]←.pc _H .sp←.sp-2; pc←nn	••••••	11010100	D4	3	3
call c,nn	if c=1 [.sp-2]←.pc _L ; [.sp-1]←.pc _H .sp←.sp-2; pc←nn	••••••	11011100	DC	3	3
call po,nn	if P _N =0 [.sp-2]←.pc _L ; [.sp-1]←.pc _H .sp←.sp-2; pc←nn	••••••	11100100	E4	3	3
call pe,nn	if P _N =1 [.sp-2]←.pc _L ; [.sp-1]←.pc _H .sp←.sp-2; pc←nn	••••••	11101100	EC	3	3
call p,nn	if s=0 [.sp-2]←.pc _L ; [.sp-1]←.pc _H .sp←.sp-2; pc←nn	••••••	11110100	F4	3	3
call m,nn	if s=1 [.sp-2]←.pc _L ; [.sp-1]←.pc _H .sp←.sp-2; pc←nn	••••••	11111100	FC	3	3
ret	.pc _L ←[.sp]; pc _H ←[.sp+1]; sp←.sp+2	••••••	11001001	C9	1	3
ret nz	if z=0 .pc _L ←[.sp]; pc _H ←[.sp+1]; sp←.sp+2	••••••	11000000	C0	1	1
ret z	if z=1 .pc _L ←[.sp]; pc _H ←[.sp+1]; sp←.sp+2	••••••	11001000	C8	1	1
ret nc	if c=0 .pc _L ←[.sp]; pc _H ←[.sp+1]; sp←.sp+2	••••••	11010000	D0	1	1
ret c	if c=1 .pc _L ←[.sp]; pc _H ←[.sp+1]; sp←.sp+2	••••••	11011000	D8	1	1
ret po	if P _N =0 .pc _L ←[.sp]; pc _H ←[.sp+1]; sp←.sp+2	••••••	11100000	E0	1	1
ret pe	if P _N =1 .pc _L ←[.sp]; pc _H ←[.sp+1]; sp←.sp+2	••••••	11101000	E8	1	1
ret p	if s=0 .pc _L ←[.sp]; pc _H ←[.sp+1]; sp←.sp+2	••••••	11110000	F0	1	1
ret m	if s=1 .pc _L ←[.sp]; pc _H ←[.sp+1]; sp←.sp+2	••••••	11111000	F8	1	1
reti	interrupt return	••••••	11101101 01001101	ED 4D	2	5
retn	Non Maskable Interrupt return	••••••	11101101 01000101	ED 45	2	5
brk k	[.sp-2]←.pc _L ; [.sp-1]←.pc _H .sp←.sp-2; pc _L ←k; pc _H ←0	••••••	11k/0111		1	4

入出力命令						
ニーモニック	命令動作	flags S Z H P _N C	オペコード		bytes	clocks
			76543210	Hex		
in .a,[n]	.a←[n]	••••••	11011011	DB	2	3
in r, _r [c]	r←[c]	↑↑↑↑00	11101101 01r000	ED	2	3
in .f,[c]	[c]	↑↑↑↑00	11101101 01110000	ED 70	2	3
in [.hl++],[c]	[.hl]←[c]; b←.b-1 .hl←.hl+1	?↑↑↑↑1 .1	11101101 10100010	ED A2	2	4
in [.hl--],[c]	[.hl]←[c]; b←.b-1 .hl←.hl-1	?↑↑↑↑1 .1	11101101 10101010	ED AA	2	4
inm [.hl++],[c]	repeat; [.hl]←[c]; b←.b-1 .hl←.hl+1; until .b=0	?↑↑↑↑1	11101101 10110010	ED B2	2	4
inm [.hl--],[c]	repeat; [.hl]←[c]; b←.b-1 .hl←.hl-1; until .b=0	?↑↑↑↑1	11101101 10111010	ED BA	2	4
out [n],a	[n]←.a	••••••	11010011	D3	2	3
out [c],r	[c]←r	••••••	11101101 01r001	ED	2	3
out [c],[hl++]	[c]←[.hl]; b←.b-1 .hl←.hl+1	?↑↑↑↑1 .1	11101101 10100011	ED A3	2	4
out [c],[hl--]	[c]←[.hl]; b←.b-1 .hl←.hl-1	?↑↑↑↑1 .1	11101101 10101011	ED AB	2	4
outm [c],[hl++]	repeat; [c]←[.hl]; b←.b-1 .hl←.hl+1; until .b=0	?↑↑↑↑1	11101101 10110011	ED B3	2	4
outm [c],[hl--]	repeat; [c]←[.hl]; b←.b-1 .hl←.hl-1; until .b=0	?↑↑↑↑1	11101101 10111011	ED BB	2	4

.1.b-1=0の時1他は0

in .f,[c] はcレジスタが示すポートの内容によってフラグを変えるだけでその内容はどこにも格納されません

CPU 制御命令						
ニーモニック	命令動作	flags S Z H P _N C	オペコード		bytes	clocks
			76543210	Hex		
adj .a	adjust to decimal	↑↑↑↑P↑	00100111	27	1	1
not .a	.a←NOT .a	••1•1•	00101111	2F	1	1
neg .a	.a←NOT .a+1	↑↑↑↑V↑	11101101 01000100	ED 44	2	2
notc	c←NOT c	••?•0↑	00111111	3F	1	1
setc	c←1	••0•0↑	00110111	37	1	1
nop	NO operation	••••••	00000000	00	1	1
halt	HALT	••••••	01110110	76	1	2
di	IFF←0	••••••	11110011	F3	1	2
ei	IFF←1	••••••	11111011	FB	1	1
im 0	interrupt mode 0	••••••	11101101 01000110	ED 46	2	3
im 1	interrupt mode 1	••••••	11101101 01010110	ED 56	2	3
im 2	interrupt mode 2	••••••	11101101 01011110	ED 5E	2	3

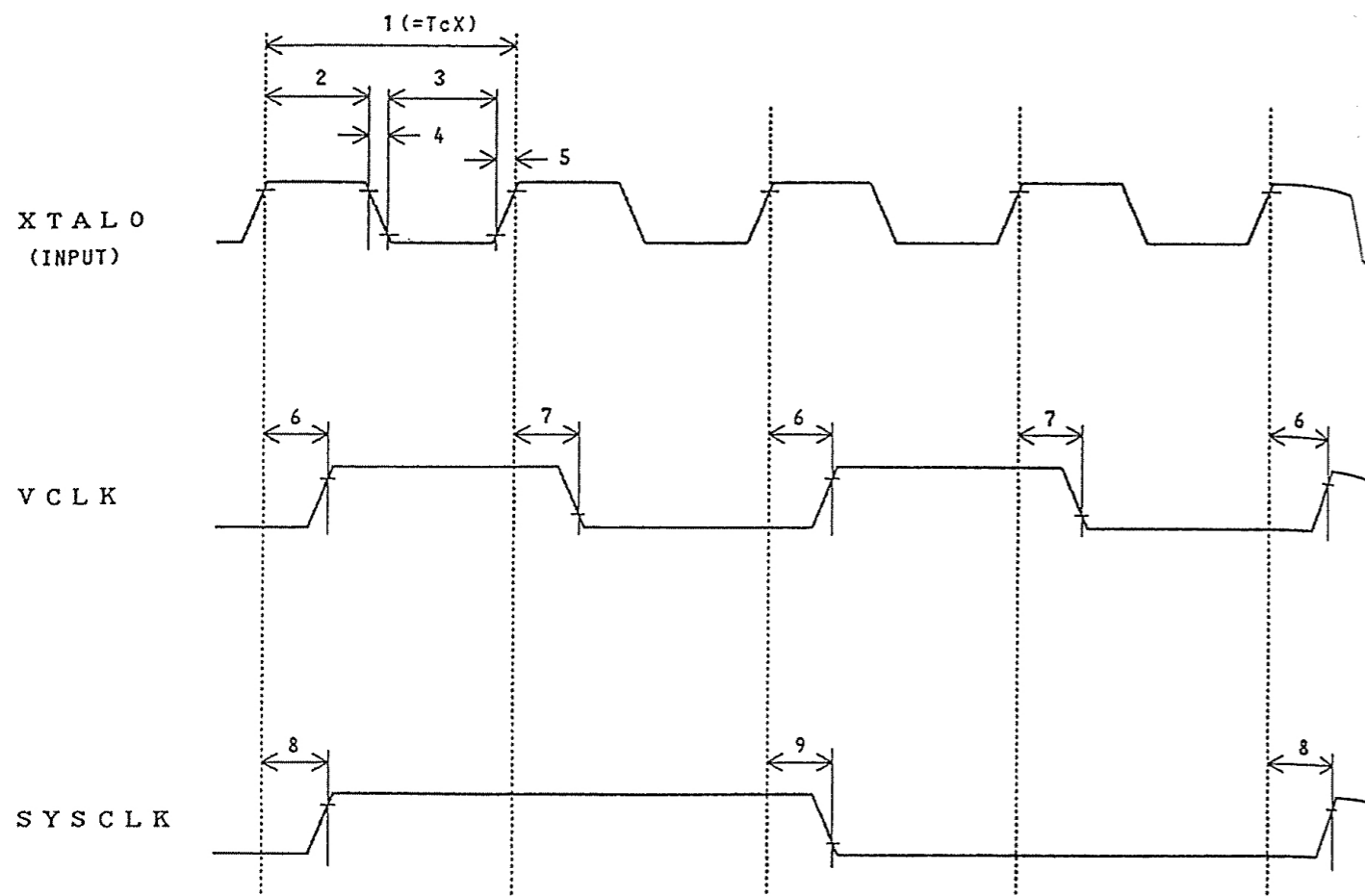


fig1 Clock Timing

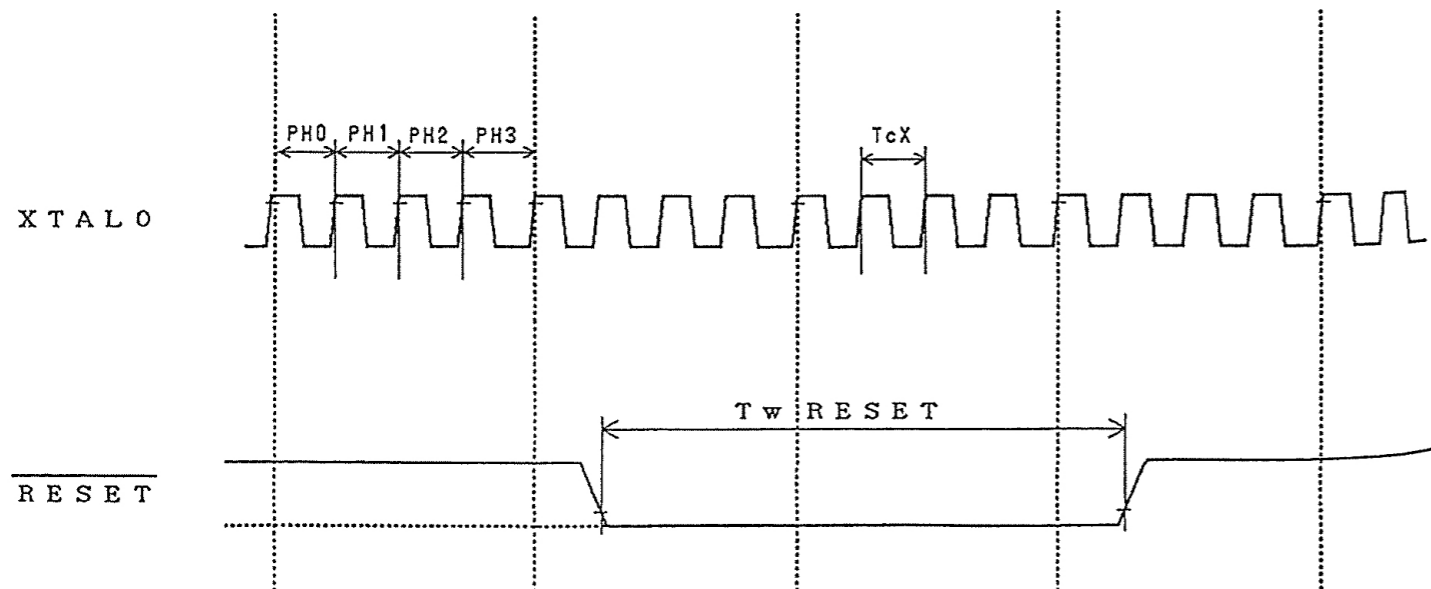
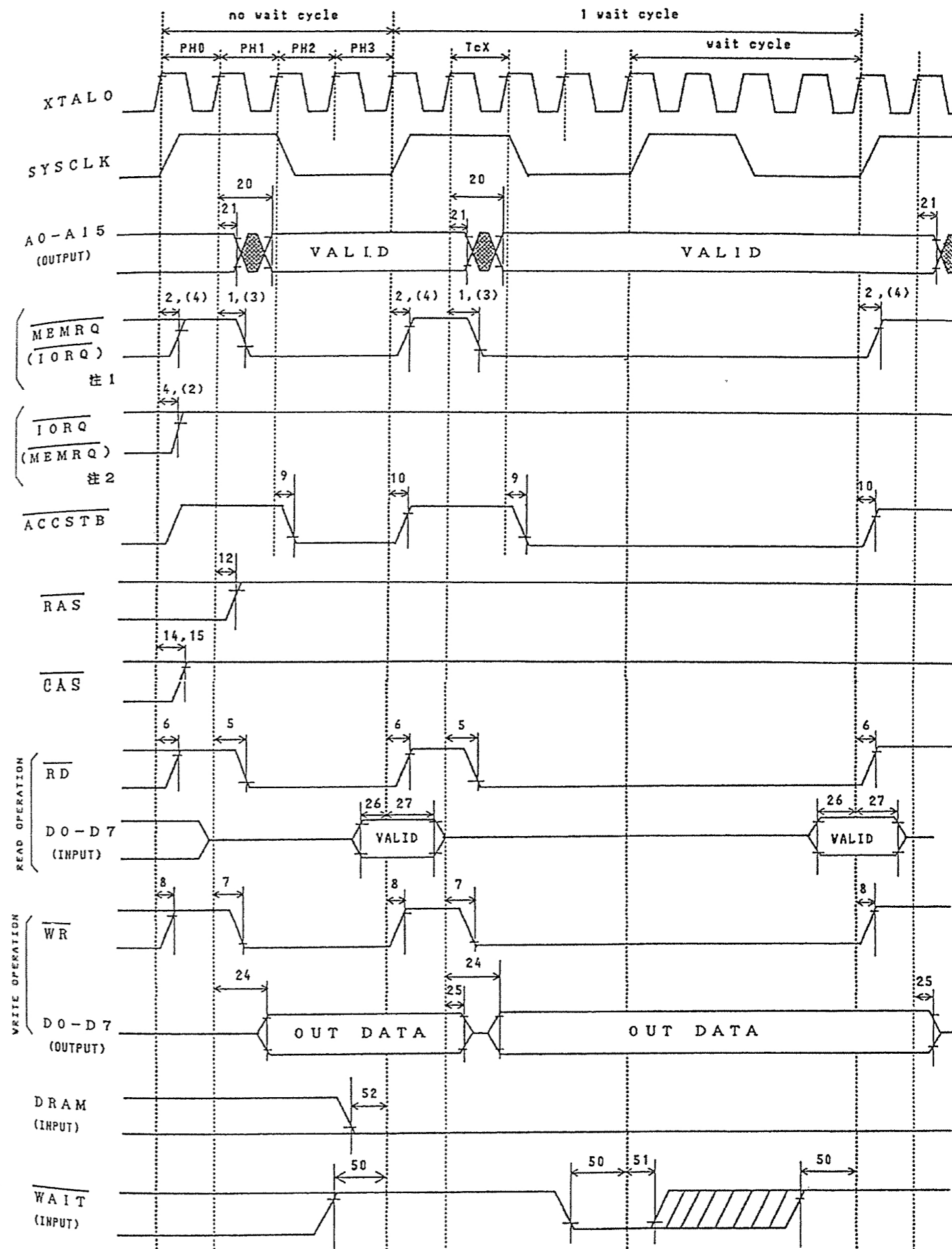


fig2 Reset Timing



注1 I/Oアクセスの場合は()を用いる
 注2 I/Oアクセス以外の場合は()を用いる

fig3-1 SRAM, ROM, I/O Read or Write Cycle

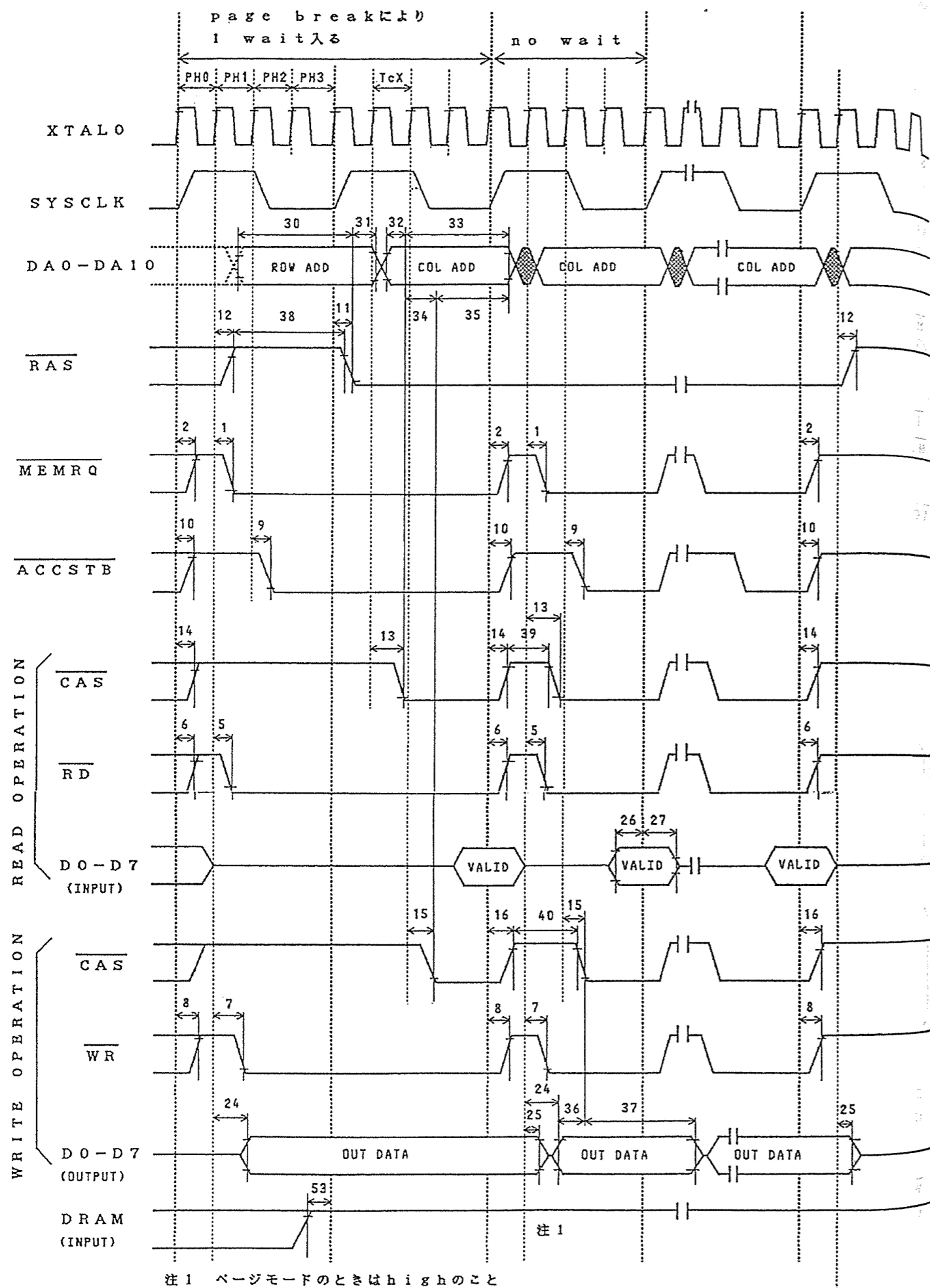


fig3-2 DRAM Read or Write Cycle

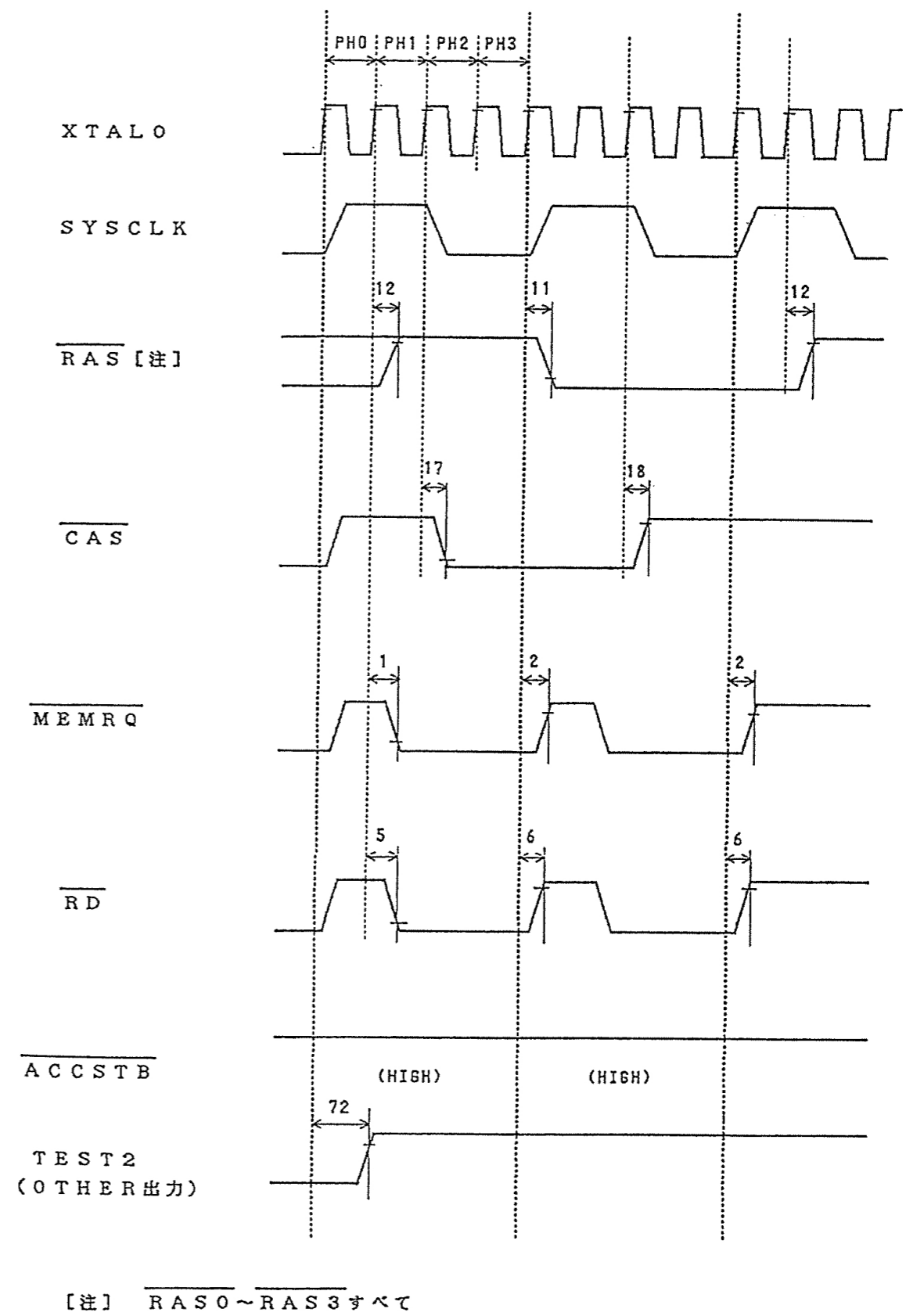
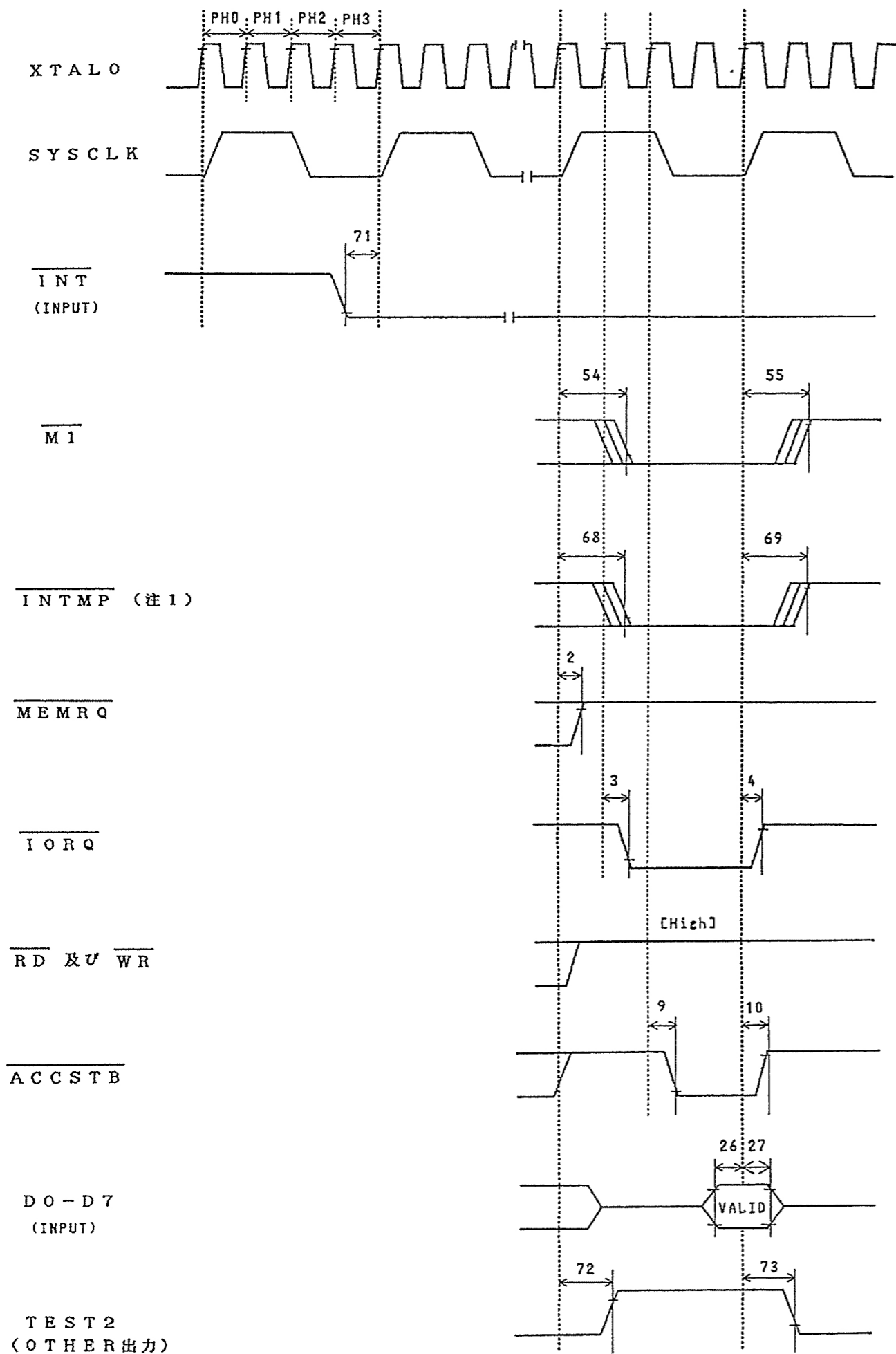


fig3-3 Refresh Cycle
(CAS Before RAS Refresh)



注1 この場合のINTMPは、mode 0の例

D0-D7 = Restart Code (mode0)
D0-D7 = Int Vector Low Address (mode2)

fig3-4 Interrupt Mode 0 or Mode 2

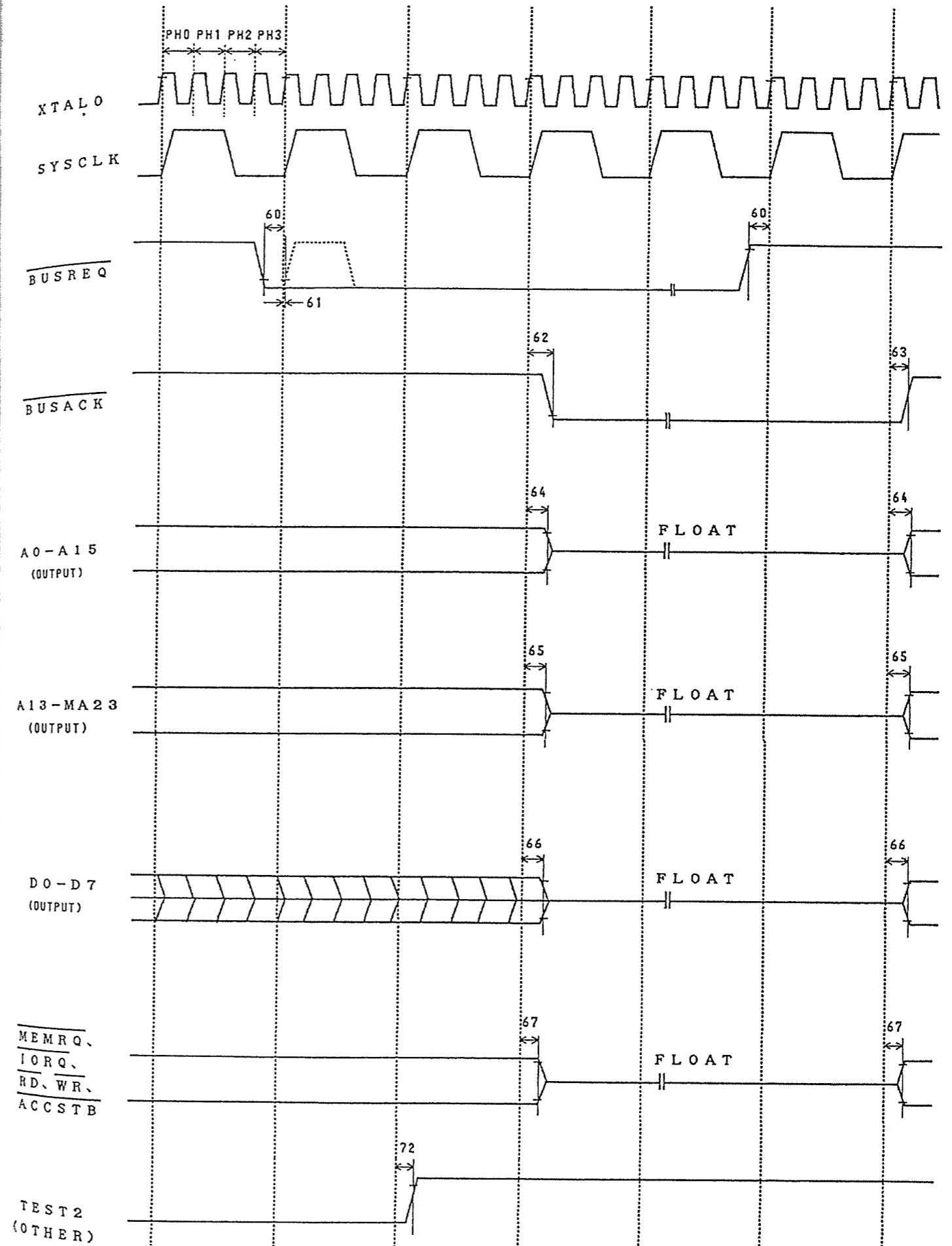
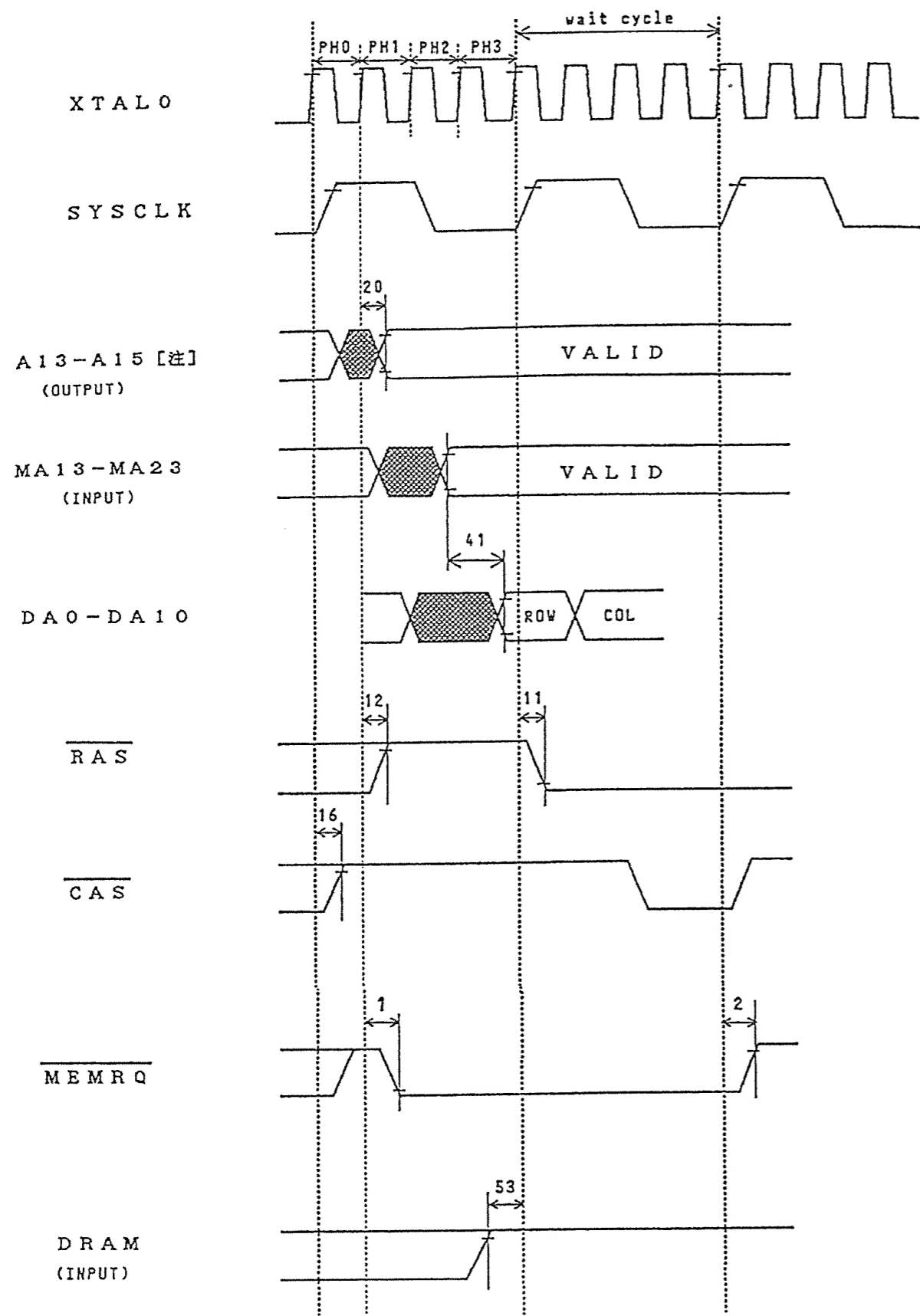


fig3-5 Bus Request/Acknowledge Cycle



入力条件: MABUSDIR = Low

【注】: A13-A15が変化するとR800内部でPAGE BREAKが発生する。
 このとき、DRAM=Highの場合はWAIT信号の値にかかわらず、
 wait cycleが1回はいる

fig3-6 External Address Mode (DRAM Accessの場合)

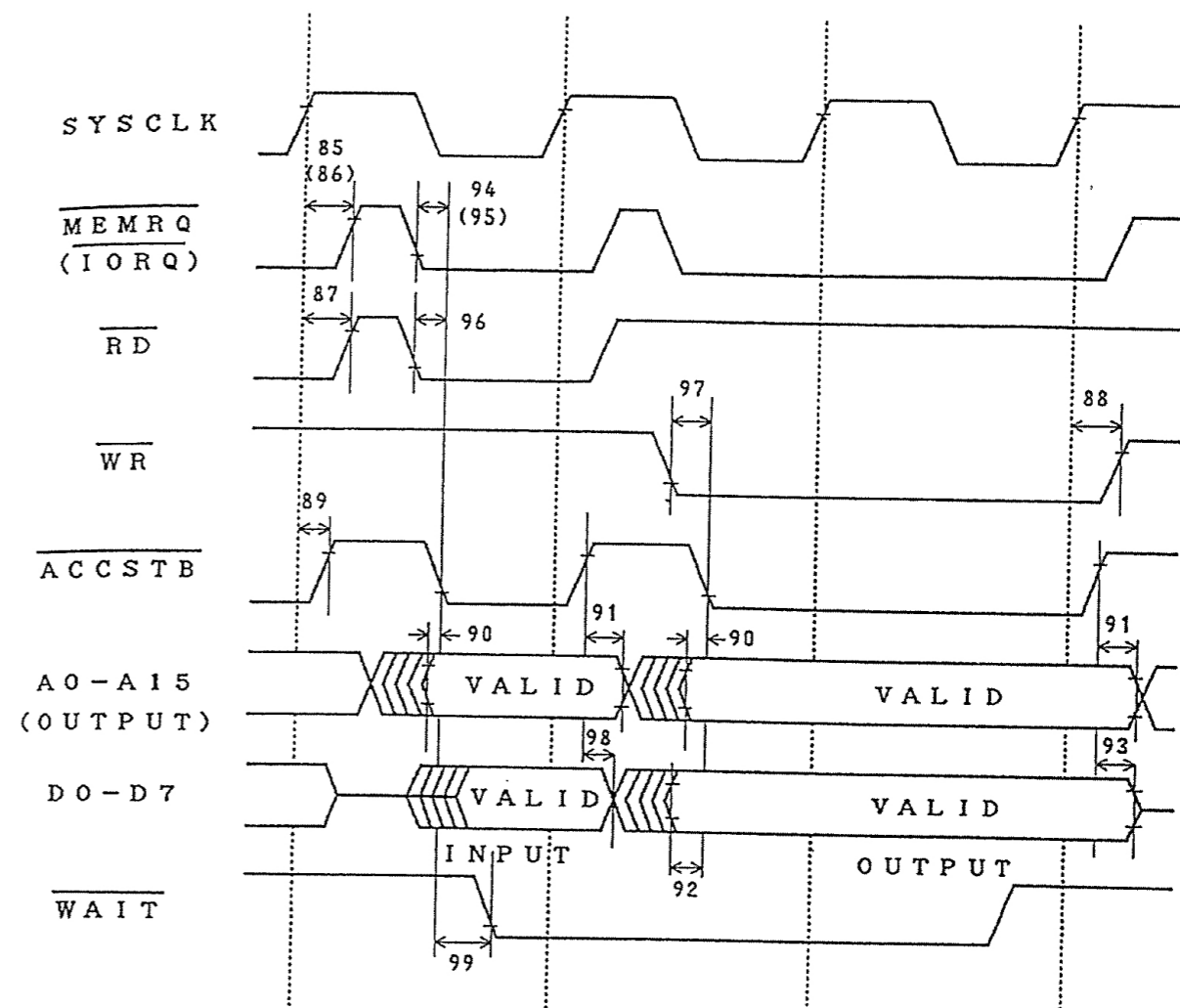


fig3-7 SYSCLK, ACCSTBと他の信号の関連 (1)

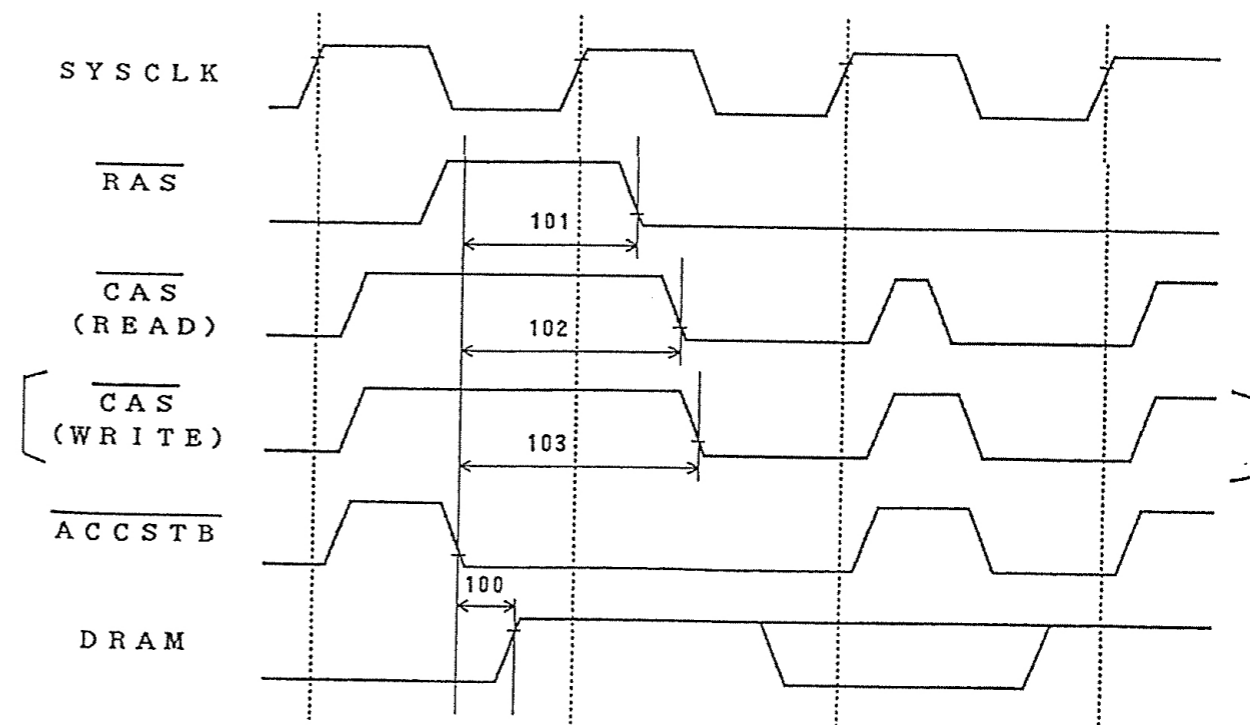


fig3-8 ACCSTBと他の信号の関連 (2)

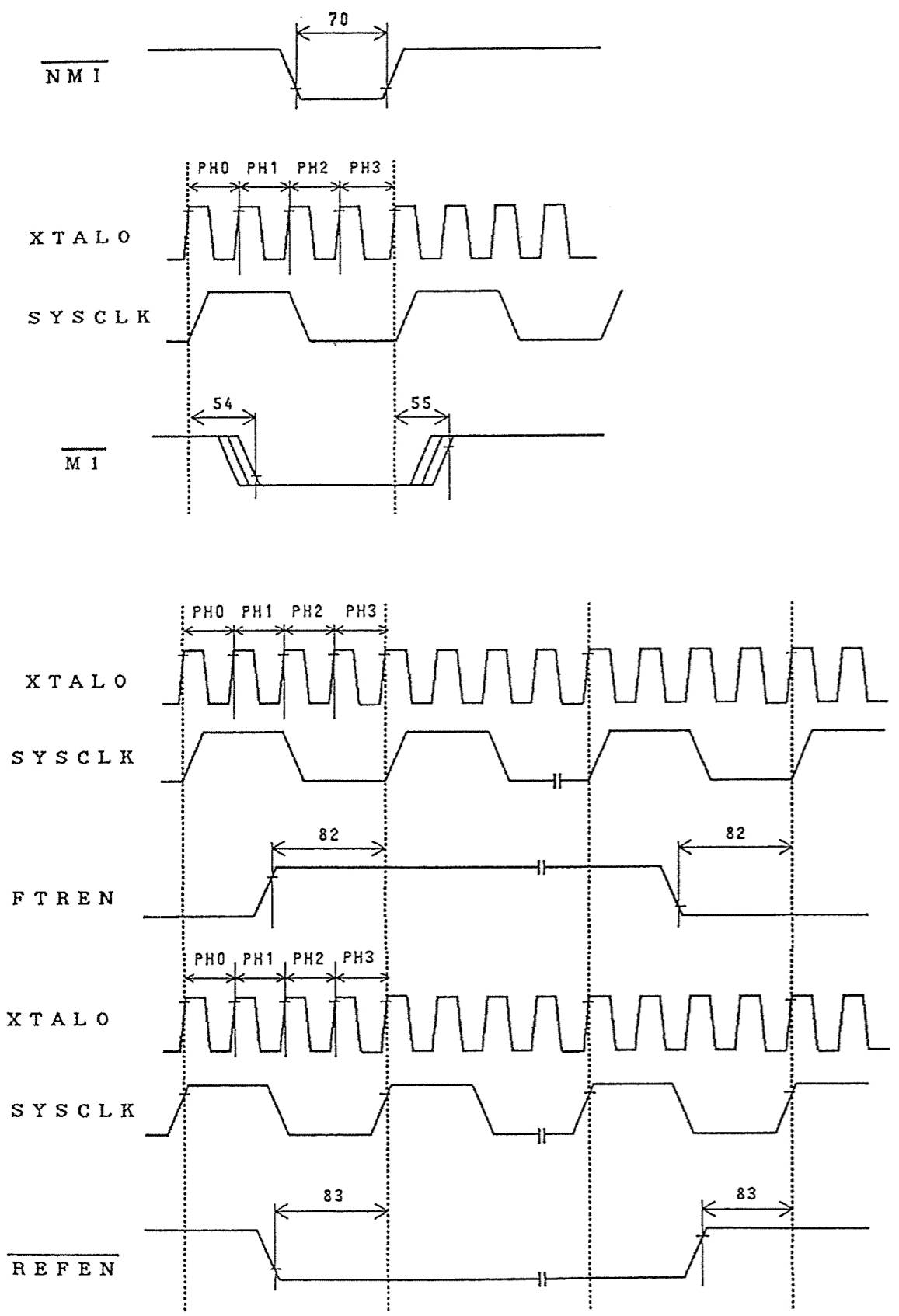


fig3-9 その他

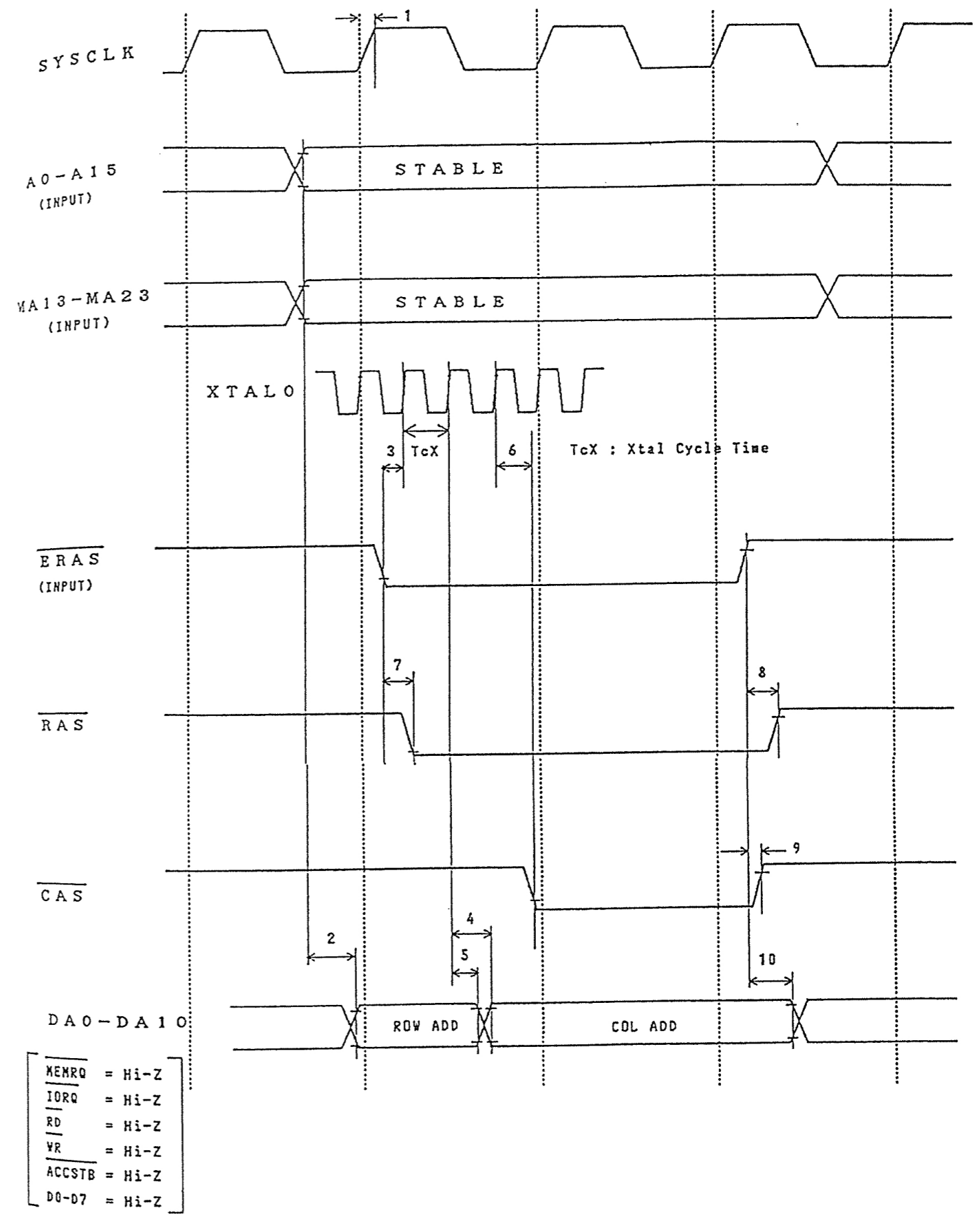


fig4 External Address Mode (BUSACK=Low, REFEN=High)