

# OKI 電子デバイス

## MSM80C88A-10

8Bit CMOS Microprocessor

### ■ 概要

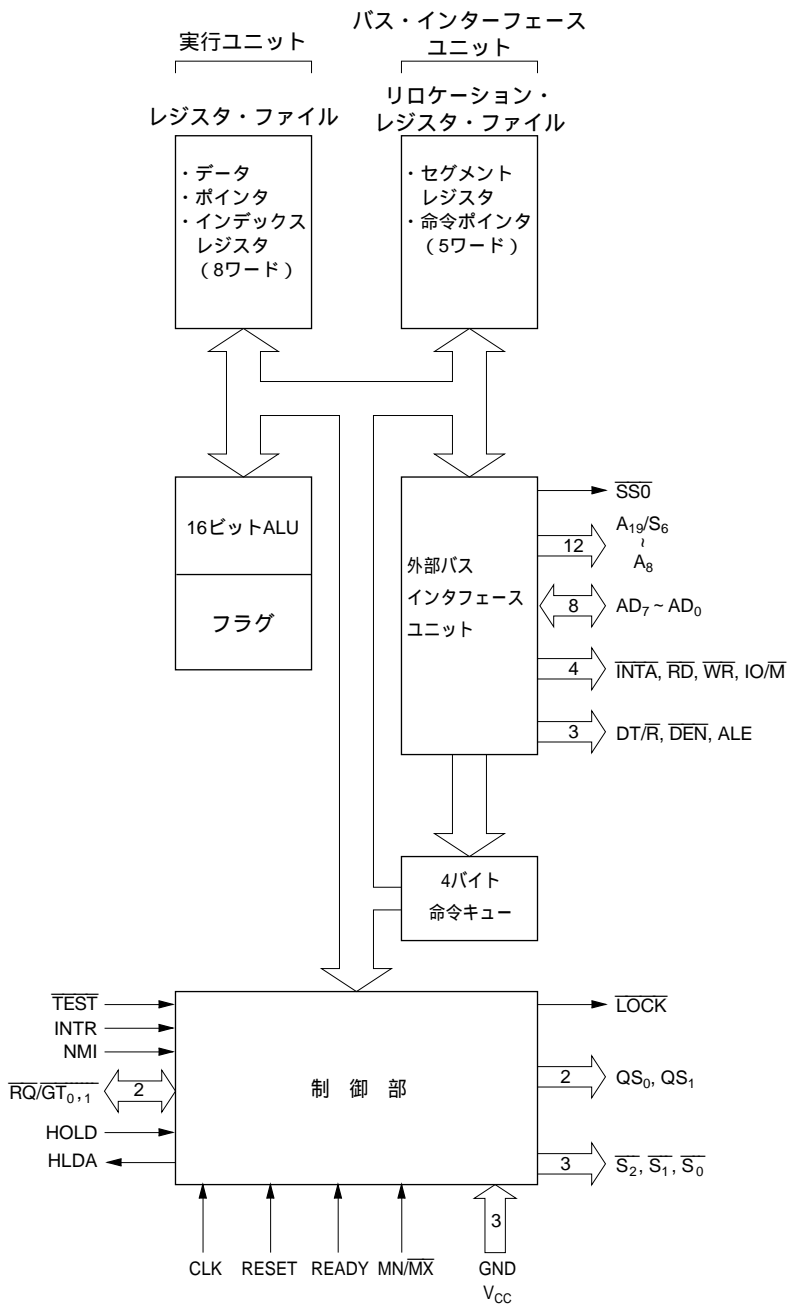
MSM80C88A-10はMSM80C88A-2のアップバージョンとしてクロックデューティを約50%とすることで10MHz動作を実現しました。また8MHz、5MHzではMSM80C88A-2、MSM80C88Aスペックを満足します。

またシリコンゲートCMOSテクノロジーを用いているので高速かつ低消費電力です。

### ■ 特長

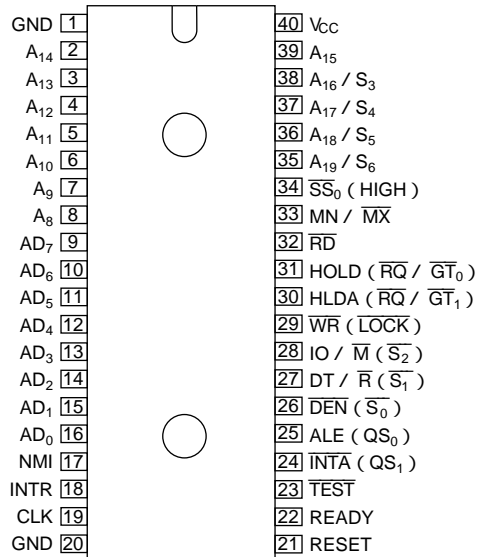
- シリコンゲートCMOSテクノロジーによる高速・低消費電力動作  
動作時 ( $I_{CC0}$ ) = 10mA / MHz  
待機時 ( $I_{CCS}$ ) = 500 $\mu$ A (最大)
- 5V 単一電源
- 0 ~ +70 ( - 40 ~ +85 ; 5MHz ) の広い動作温度保証範囲
- 完全スタティック動作 (動作周波数 ; DC ~ 10MHz)
- バスホールド回路採用によりプルアップ抵抗不用
- 1メガ・バイトのメモリ空間に直接アドレス可能
- 高級言語のデータ構造を効率よく実現できる強力なアセンブリ言語  
アドレッシング・モードは6タイプ24種類  
強力なストリング動作とループ動作
- 14ワードの16ビット・レジスタ
- ビット、バイト、ワードおよびブロック操作可能
- 乗除算を含む2進 / 10進表現の“符号付”および“符号なし”算術演算が可能
- 40ピンプラスチックDIP ( DIP40-P-600-2.54 ) : ( 製品名 : MSM80C88A-10RS )
- 44ピンプラスチックQFJ ( QFJ44-P-S650-1.27 ) : ( 製品名 : MSM80C88A-10JS )
- 56ピンプラスチックQFP ( QFP56-P-1519-1.00-K ) : ( 製品名 : MSM80C88A-10GS-K )

## ■ 回路構成



## ■ 端子接続（上面図）

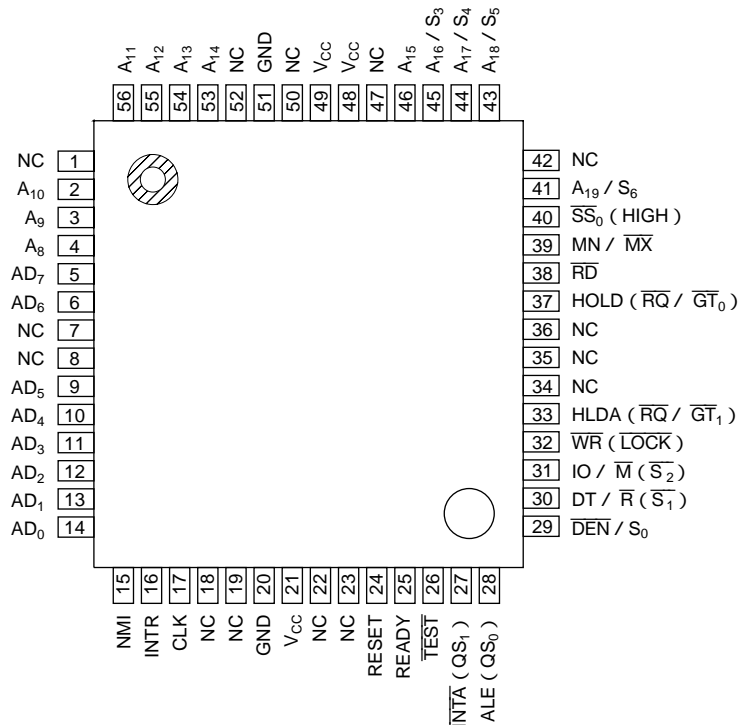
40ピンプラスチックDIP MSM80C88A-10RS



注記1： マキシマム・モードの端子接続は、( ) 内に示しました。

注記2： 形名の現品への表示は簡略形を用い、M80C88A-10 となります。

56ピンプラスチックQFP MSM80C88A-10GS-K

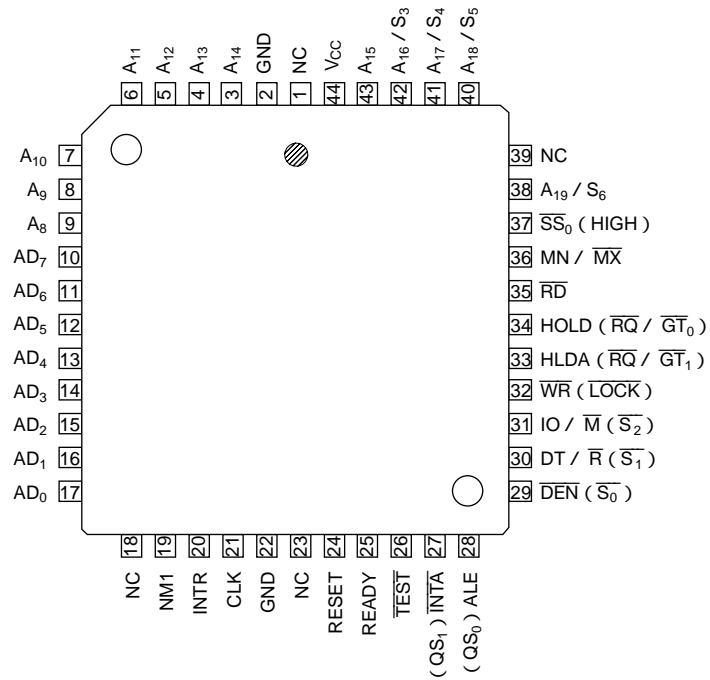


注記1： NCは未使用端子。

注記2： マキシマム・モードの端子接続は、( ) 内に示しました。

注記3： 形名の現品への表示は簡略形を用い、M80C88A-10 となります。

## 44ピンプラスチックQFJ MSM80C88A-10JS



注記1： NCは未使用端子。

注記2： マキシマム・モードの端子接続は、( )内に示しました。

注記3： 形名の現品への表示は簡略形を用い、M80C88A-10 となります。

## ■ 電気的特性

## ● 絶対最大定格

項目	記号	条件	規格値		単位
			MSM80C88A-10RS	MSM80C88A-10GS MSM80C88A-10JS	
電源電圧	$V_{CC}$	GNDを基準にした場合	- 0.5 ~ + 7.0		V
入力電圧	$V_{IN}$		- 0.5 ~ $V_{CC} + 0.5$		V
出力電圧	$V_{OUT}$		- 0.5 ~ $V_{CC} + 0.5$		V
保存温度	$T_{STG}$		- 65 ~ + 150		
許容損失	$P_D$	$T_a = 25$	1.0	0.7	W

## ● 動作範囲

項目	記号	範囲	単位
電源電圧	$V_{CC}$	4.5 ~ 5.5	V
動作温度	$T_{OP}$	- 40 ~ + 85	

## ● 推奨動作条件

項目	記号	Min.	Typ.	Max.	備考	単位
電源電圧	$V_{CC}$	4.75	5.0	5.25		V
動作温度	$T_{OP}$	0	25	70		
"L"入力電圧	$V_{IL}$	- 0.5		0.8		V
	$V_{CL}$	- 0.5		0.8	CLJ	
"H"入力電圧	$V_{IH}$	2.0		$V_{CC} + 0.5$		V
	$V_{CH}$	$V_{CC} - 0.8$		$V_{CC} + 0.5$	CLK	

## ● 直流特性

Ta = - 40 ~ + 85 , V<sub>CC</sub> = 4.5V ~ 5.5V

項目	記号	条件	Min.	Max.	単位
"L"出力電圧	V <sub>OL</sub>	I <sub>LO</sub> = 2.5mA		0.4	V
"H"出力電圧	V <sub>OH</sub>	I <sub>LO</sub> = - 2.5mA	3.0		V
		I <sub>LO</sub> = - 100μA	V <sub>CC</sub> - 0.4		V
入力リーク電流	I <sub>LI</sub>	0 V <sub>IN</sub> V <sub>CC</sub>	- 1.0	1.0	μA
出力リーク電流	I <sub>LO</sub>	V <sub>OUT</sub> = V <sub>CC</sub> または, GND	- 10	10	μA
出力リーク電流 (バスホールド"L")	I <sub>BHL</sub>	V <sub>IN</sub> = 0.8V 注1	50	400	μA
入力リーク電流 (バスホールド"H")	I <sub>BHH</sub>	V <sub>IN</sub> = 3.0V 注2	- 50	- 400	μA
バスホールド反転電流 ("L" → "H") 注3	I <sub>BHLO</sub>			600	μA
バスホールド反転電流 ("H" → "L") 注4	I <sub>BHHO</sub>		-	- 600	μA
動作時電源電流	I <sub>CCS</sub>	V <sub>IL</sub> = GND, V <sub>IH</sub> = V <sub>CC</sub>		10	mA/MHz
待機時電源電流	I <sub>CC</sub>	V <sub>IN</sub> = V <sub>CC</sub> または, GND CLK = V <sub>CC</sub> または, GND 出力は, 無負荷		500	μA
入力容量	C <sub>IN</sub>	・動作周波数 = 1MHz ・被測定端子以外の 端子はGNDに固定 ・V <sub>IN</sub> = 5.0V または, GND		10	pF
出力容量	C <sub>OUT</sub>			15	pF
入出力容量	C <sub>IO</sub>			20	pF

注記1 : V<sub>IN</sub> = GNDにした後、V<sub>IN</sub> = 0.8Vにする (2 ~ 16、35 ~ 39ピン)

注記2 : V<sub>IN</sub> = V<sub>CC</sub>にした後、V<sub>IN</sub> = 3.0Vにする (2 ~ 16、26 ~ 32、34 ~ 39ピン)

注記3 : “L”レベルから“H”レベルに反転するときには、ピンに対し少なくともI<sub>BHLO</sub>の電流を流し込む必要がある (2 ~ 16、35 ~ 39ピン)

注記4 : “H”レベルから“L”レベルに反転するときには、ピンに対し少なくともI<sub>BHHO</sub>の電流を引き出す必要がある (2 ~ 16、26 ~ 32、34 ~ 39ピン)

注記5 : I<sub>BHL</sub> / I<sub>BHH</sub>対象ピンにはI<sub>LO</sub> / I<sub>LI</sub>規格は適用されません。

## ● 交流特性

(1) ミニマムモード時  
タイミング特性

項目	記号	5MHz Spec. V <sub>CC</sub> = 4.5V ~ 5.5V Ta = -40 ~ +85		8MHz Spec. V <sub>CC</sub> = 4.75V ~ 5.25V Ta = 0 ~ +70		10MHz Spec. V <sub>CC</sub> = 4.75V ~ 5.25V Ta = 0 ~ +70		単位
		Min.	Max.	Min.	Max.	Min.	Max.	
CLK周期	t <sub>CLCL</sub>	200	DC	125	DC	100	DC	ns
CLK "L"レベル時間	t <sub>CLCH</sub>	118		68		46		ns
CLK "H"レベル時間	t <sub>CHC</sub>	69		44		44		ns
CLK立ち上がり時間 (1.0V → 3.5V)	t <sub>CH1CH2</sub>		10		10		10	ns
CLK立ち下り時間 (3.5V → 1.0V)	t <sub>CL2CL1</sub>		10		10		10	ns
データ入力セットアップ時間	t <sub>DVCL</sub>	30		20		20		ns
データ入力ホールド時間	t <sub>CLDX</sub>	10		10		10		ns
RDYセットアップ時間 注1, 注2	t <sub>RIVCL</sub>	35		35		35		ns
RDYホールド時間 注1, 注2	t <sub>CLRIX</sub>	0		0		0		ns
READYアクティブセット 時間	t <sub>RYHCH</sub>	118		68		46		ns
READYホールド時間	t <sub>CHRYX</sub>	30		20		20		ns
READYインアクティブ セットアップ時間 注3	t <sub>RYLCL</sub>	- 8		- 8		- 8		ns
HOLDセットアップ時間	t <sub>HVCH</sub>	35		20		20		ns
INTR, NMI, $\overline{\text{TEST}}$ セットアップ時間 注2	t <sub>INVCH</sub>	30		15		15		ns
入力立ち上がり時間 (CLKを 除く) (0.8V → 2.0V)	t <sub>LIH</sub>		15		15		15	ns
入力立ち下り時間 (CLKを 除く) (2.0V → 0.8V)	t <sub>HIL</sub>		15		15		15	ns

注記1： 参考のため、MSM82C84Aの信号を示しました。(規格値は、最新のデータシートで確認して下さい。)

注記2： 次のクロックで取り込まれるのに必要なセットアップ時間です。

注記3： T<sub>2</sub>ステートだけに適用されます。(T<sub>3</sub>ステート内に8ns)

## スイッチング特性

(負荷容量 = 20 ~ 100pF)

項目	記号	5MHz Spec. V <sub>CC</sub> = 4.5V ~ 5.5V Ta = -40 ~ +85		8MHz Spec. V <sub>CC</sub> = 4.75V ~ 5.25V Ta = 0 ~ +70		10MHz Spec. V <sub>CC</sub> = 4.75V ~ 5.25V Ta = 0 ~ +70		単位
		Min.	Max.	Min.	Max.	Min.	Max.	
アドレスバリッド遅延時間	t <sub>CLAV</sub>	10	110	10	60	10	60	ns
アドレスホールド時間	t <sub>CLAX</sub>	10		10		10		ns
アドレスフロート遅延時間	t <sub>CLAZ</sub>	t <sub>CLAX</sub>	80	t <sub>CLAX</sub>	50	t <sub>CLAX</sub>	50	ns
ALEパルス幅	t <sub>LHLL</sub>	t <sub>CLCH-20</sub>		t <sub>CLCH-10</sub>		t <sub>CLCH-10</sub>		ns
ALEアクティブ遅延時間	t <sub>CLLH</sub>		80		50		40	ns
ALEインアクティブ遅延時間	t <sub>CHLL</sub>		85		55		45	ns
ALEインアクティブに対する アドレスホールド時間	t <sub>LLAX</sub>	t <sub>CHCL-10</sub>		t <sub>CHCL-10</sub>		t <sub>CHCL-10</sub>		ns
データバリッド遅延時間	t <sub>CLDV</sub>	10	110	10	60	10	60	ns
データホールド時間	t <sub>CHDX</sub>	10		10		10		ns
WRインアクティブ後データ ホールド時間	t <sub>WHDX</sub>	t <sub>CLCH-30</sub>		t <sub>CLCH-30</sub>		t <sub>CLCH-25</sub>		ns
制御信号アクティブ遅延時間 (1)	t <sub>CVCTV</sub>	10	110	10	70	10	55	ns
制御信号アクティブ遅延時間 (2)	t <sub>CHCTV</sub>	10	110	10	60	10	50	ns
制御信号インアクティブ 遅延時間	t <sub>CVCTX</sub>	10	110	10	70	10	55	ns
アドレスフロート後RDアク ティブ時間	t <sub>AZRL</sub>	0		0		0		ns
RDアクティブ遅延時間	t <sub>CLRL</sub>	10	165	10	100	10	70	ns
RDインアクティブ遅延時間	t <sub>CLRH</sub>	10	150	10	80	10	60	ns
RDインアクティブ後の 次アドレスアクティブ時間	t <sub>RHAV</sub>	t <sub>CLCL-45</sub>		t <sub>CLCL-40</sub>		t <sub>CLCL-35</sub>		ns
HLDAバリッド遅延時間	t <sub>CLHAV</sub>	10	160	10	100	10	60	ns
RDパルス幅	t <sub>RLRH</sub>	2t <sub>CLCL-75</sub>		2t <sub>CLCL-50</sub>		2t <sub>CLCL-40</sub>		ns
WRパルス幅	t <sub>WLWH</sub>	2t <sub>CLCL-60</sub>		2t <sub>CLCL-40</sub>		2t <sub>CLCL-35</sub>		ns
ALE "L"レベルまでのアドレ スバリッド時間	t <sub>AVAL</sub>	t <sub>CLCH-60</sub>		t <sub>CLCH-40</sub>		t <sub>CLCH-35</sub>		ns
出力立ち上り時間 (0.8V → 2.0V)	t <sub>OLOH</sub>		15		15		15	ns
出力立ち下り時間 (2.0V → 0.8V)	t <sub>OHOL</sub>		15		15		15	ns

(2) マキシマムモード時 【MSM82C88-2使用】  
タイミング特性

項目	記号	5MHz Spec. V <sub>CC</sub> = 4.5V ~ 5.5V Ta = -40 ~ +85		8MHz Spec. V <sub>CC</sub> = 4.75V ~ 5.25V Ta = 0 ~ +70		10MHz Spec. V <sub>CC</sub> = 4.75V ~ 5.25V Ta = 0 ~ +70		単位
		Min.	Max.	Min.	Max.	Min.	Max.	
CLK周期	t <sub>CLCL</sub>	200	DC	125	DC	100	DC	ns
CLK "L"レベル時間	t <sub>CLCH</sub>	118		68		46		ns
CLK "H"レベル時間	t <sub>CKCL</sub>	69		44		44		ns
CLK立ち上り時間 (1.0V → 3.5V)	t <sub>CH1CK2</sub>		10		10		10	ns
CLK立ち下り時間 (3.5V → 1.0V)	t <sub>CL2CL1</sub>		10		10		10	ns
データ入力セットアップ時間	t <sub>DVCL</sub>	30		20		20		ns
データ入力ホールド時間	t <sub>CLDX</sub>	10		10		10		ns
RDYセットアップ時間 注1, 注2	t <sub>R1VCL</sub>	35		35		35		ns
RDYホールド時間 注1, 注2	t <sub>CLRIX</sub>	0		0		0		ns
READYアクティブセット アップ時間	t <sub>RYHCH</sub>	118		68		46		ns
READYホールド時間	t <sub>CHRYX</sub>	30		20		20		ns
READYインアクティブ セットアップ時間 注4	t <sub>RYLCL</sub>	- 8		- 8		- 8		ns
INTR, NMI, $\overline{\text{TEST}}$ セットアップ時間 注2	t <sub>INVCH</sub>	30		15		15		ns
$\overline{\text{RQ}} / \overline{\text{GT}}$ セットアップ時間	t <sub>GVCH</sub>	30		15		15		ns
$\overline{\text{RQ}} / \overline{\text{GT}}$ ホールド時間	t <sub>CHGX</sub>	40		30		20		ns
入力立ち上り時間 (CLKを 除く) (0.8V → 2.0V)	t <sub>LIH</sub>		15		15		15	ns
入力立ち下り時間 (CLKを 除く) (2.0V → 0.8V)	t <sub>HIL</sub>		15		15		15	ns

注記1: 参考のため、MSM82C84AおよびMSM82C88-2の信号を示しました。  
(規格値は、最新のデータシートで確認して下さい。)

注記2: 次のクロックで取り込まれるのに必要なセットアップ時間です。

注記3: T<sub>3</sub>および、T<sub>W</sub>状態だけに適用されます。

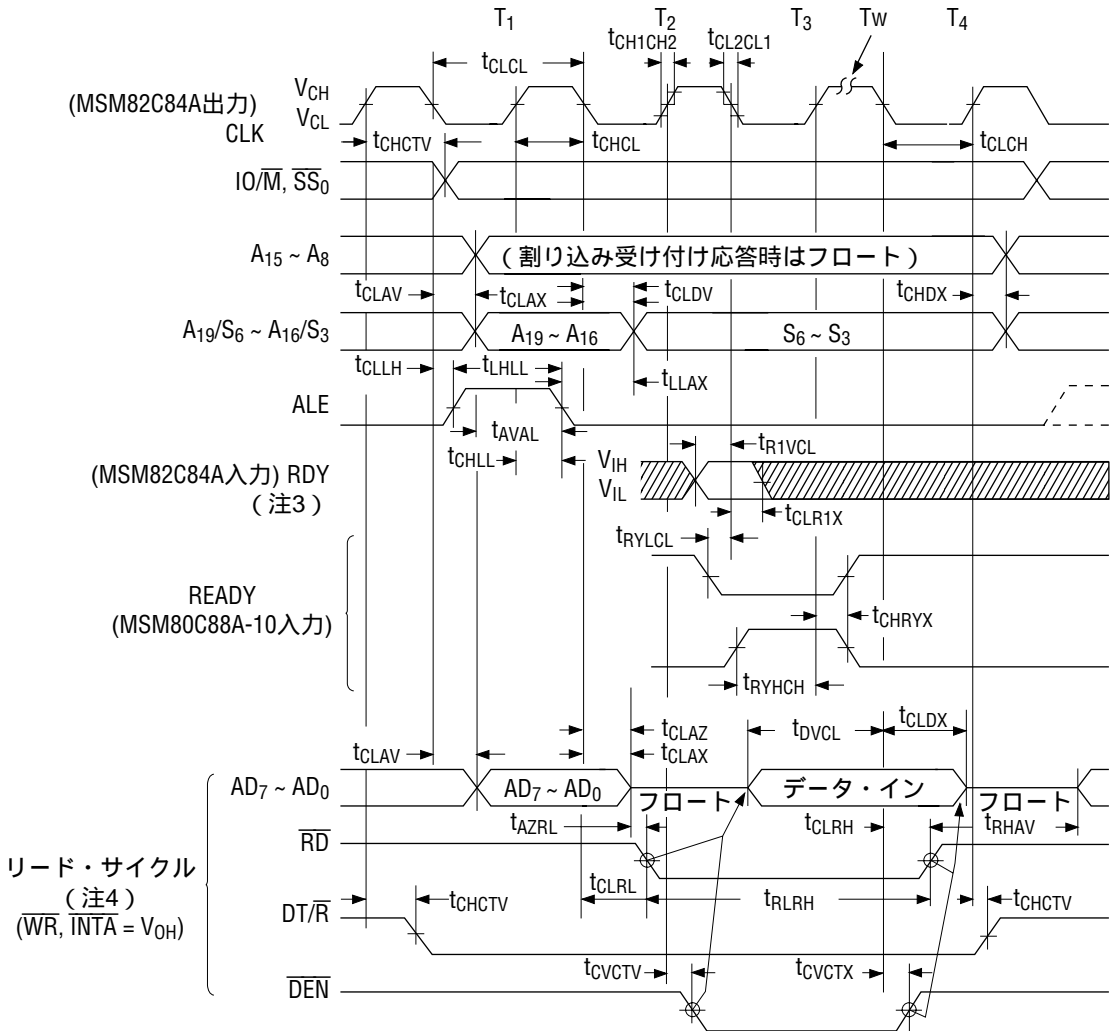
注記4: T<sub>2</sub>状態だけに適用されます。

## スイッチング特性

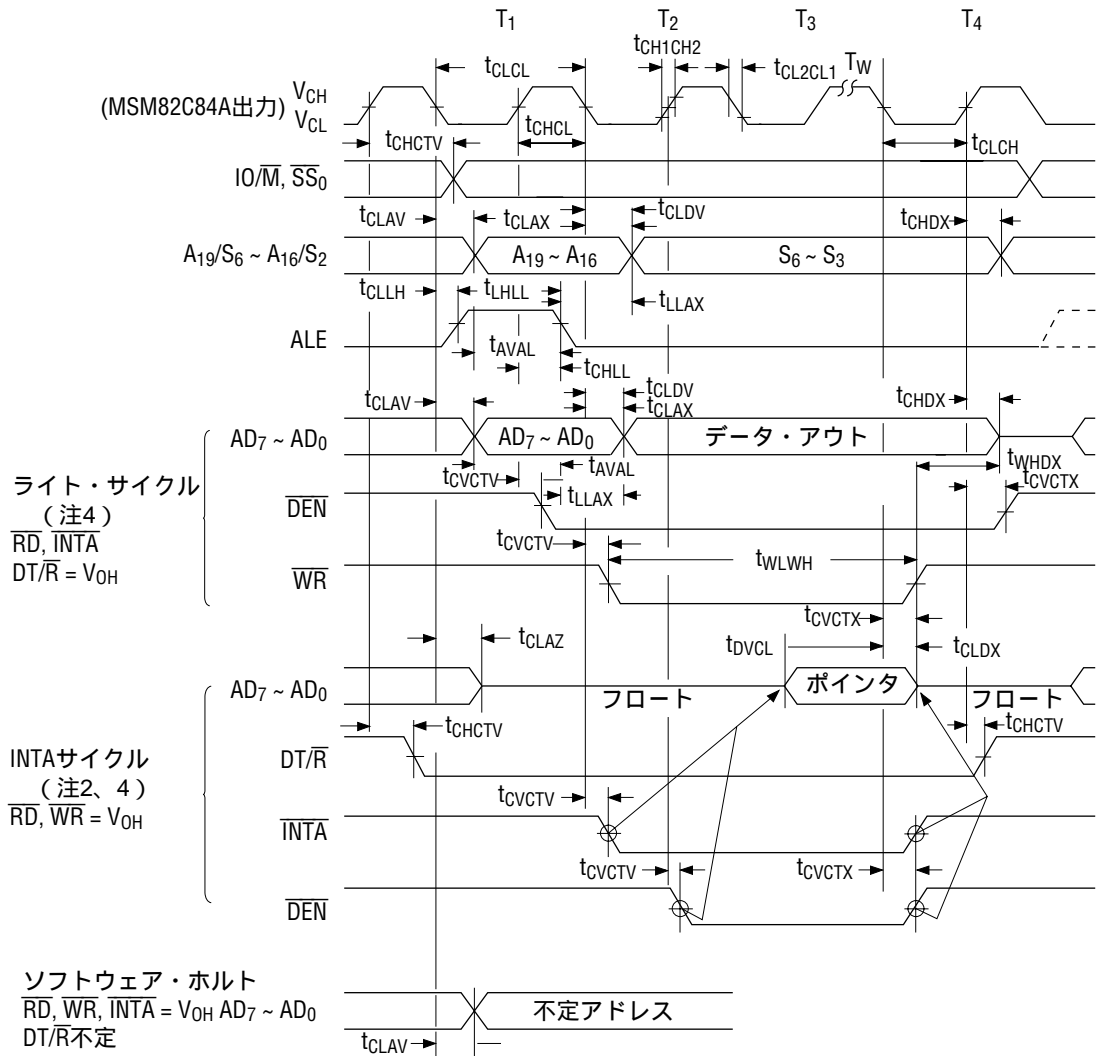
(負荷容量 = 20 ~ 100pF)

項目	記号	5MHz Spec. V <sub>CC</sub> = 4.5V ~ 5.5V Ta = -40 ~ +85		8MHz Spec. V <sub>CC</sub> = 4.75V ~ 5.25V Ta = 0 ~ +70		10MHz Spec. V <sub>CC</sub> = 4.75V ~ 5.25V Ta = 0 ~ +70		単位
		Min.	Max.	Min.	Max.	Min.	Max.	
コマンドアクティブ遅延時間 注1	t <sub>CLML</sub>	5	45	5	35	5	35	ns
コマンドインアクティブ 遅延時間 注1	t <sub>CLMH</sub>	5	45	5	45	5	45	ns
READYアクティブからのス テータス無効遅延時間 注3	t <sub>RYHSH</sub>		110		65		45	ns
ステータスアクティブ遅延 時間	t <sub>CHSV</sub>	10	110	10	60	10	45	ns
ステータスインアクティブ 遅延時間	t <sub>CLSH</sub>	10	130	10	70	10	60	ns
アドレスバリッド遅延時間	t <sub>CLAV</sub>	10	110	10	60	10	60	ns
アドレスホールド時間	t <sub>CLAX</sub>	10		10		10		ns
アドレスフロート遅延時間	t <sub>CLAZ</sub>	t <sub>CLAX</sub>	80	t <sub>CLAX</sub>	50	t <sub>CLAX</sub>	50	ns
ステータスバリッドからALE "H"レベルまでの遅延時間 注1	t <sub>SVLH</sub>		35		25		25	ns
ステータスバリッドからMCE "H"レベルまでの遅延時間 注1	t <sub>SVMCH</sub>		35		30		30	ns
CLK立ち下りからALE "H" レベルまでの遅延時間 注1	t <sub>CLLH</sub>		35		25		25	ns
CLK立ち下りからMCE "H" レベルまでの遅延時間 注1	t <sub>CLMCH</sub>		35		25		25	ns
ALEインアクティブ 遅延時間 注1	t <sub>CHLL</sub>	4	35	4	25	4	25	ns
データバリッド遅延時間	t <sub>CLDV</sub>	10	110	10	60	10	60	ns
データホールド時間	t <sub>CHDX</sub>	10		10		10		ns
制御信号アクティブ遅延時間 注1	t <sub>CVNV</sub>	5	45	5	45	5	45	ns
制御信号インアクティブ 遅延時間 注1	t <sub>CVNX</sub>	5	45	5	45	5	45	ns
アドレスフロート後RDアク ティブ時間	t <sub>AZRL</sub>	0		0		0		ns
RDアクティブ遅延時間	t <sub>CLRL</sub>	10	165	10	100	10	70	ns
RDインアクティブ遅延時間	t <sub>CLRH</sub>	10	150	10	80	10	60	ns
RDインアクティブ後の 次アドレスアクティブ時間	t <sub>RHAV</sub>	t <sub>CLCL-45</sub>		t <sub>CLCL-40</sub>		t <sub>CLCL-35</sub>		ns
DT / R̄アクティブ遅延時間 注1	t <sub>CHDTL</sub>		50		50		50	ns
DT / R̄インアティブ遅延時間 注1	t <sub>CHDTH</sub>		35		30		30	ns
GT̄アクティブ遅延時間	t <sub>CLGL</sub>	0	85	0	50	0	45	ns
GT̄インアクティブ遅延時間	t <sub>CLGH</sub>	0	85	0	50	0	45	ns
RDパルス幅	t <sub>RLRH</sub>	2t <sub>CLCL-75</sub>		2t <sub>CLCL-50</sub>		2t <sub>CLCL-40</sub>		ns
出力立ち上り時間 (0.8V → 2.0V)	t <sub>OLOH</sub>		15		15		15	ns
出力立ち下り時間 (2.0V → 0.8V)	t <sub>OHOL</sub>		15		15		15	ns

■ タイミング図



ミニマムモード(1)



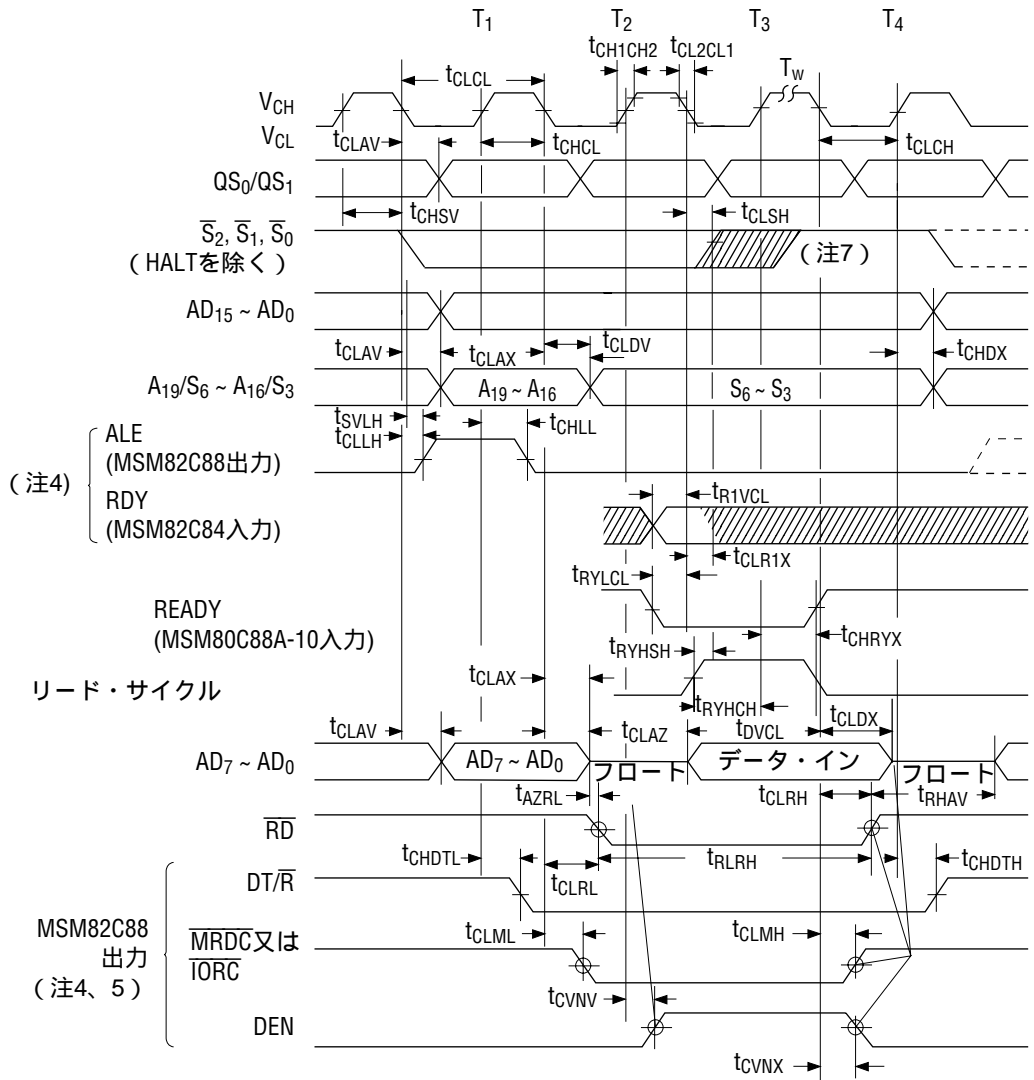
注記1: RDY 信号は T<sub>2</sub>、T<sub>3</sub>、T<sub>W</sub> ステートの終わりでサンプルされ、必要に応じて T<sub>W</sub> ステートを挿入します。

注記2: 2つのINTAサイクルがあり、2番目のINTAサイクルではMSM80C88A-10のローカル・アドレス/データバスはフローティング状態となります。なお、ポインタ・アドレスに関する制御信号は、2番目のINTAサイクルについて示してあります。

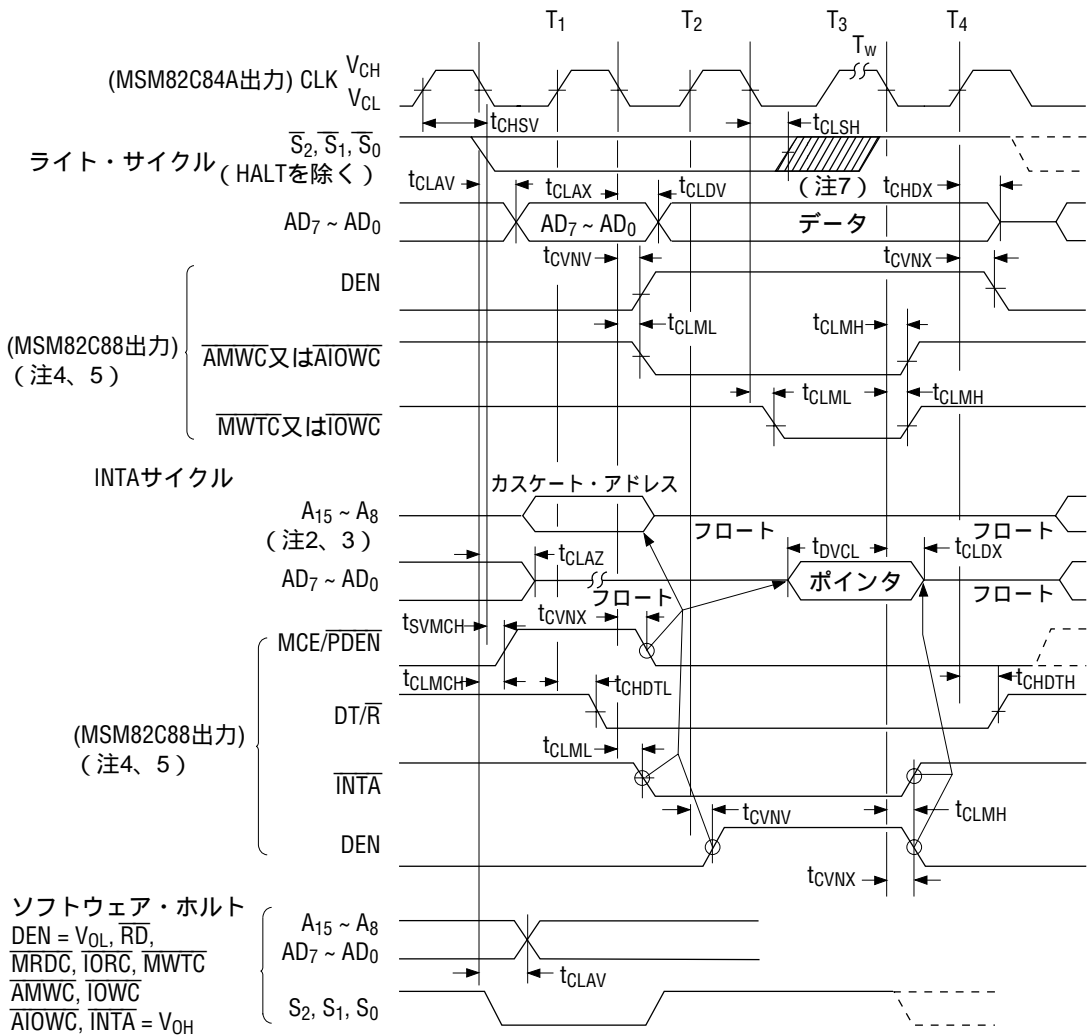
注記3: MSM82C84A (クロック・ジェネレータ) の信号は参考として示しました。

注記4: 特に示されている場合を除いて、すべてのタイミングは1.5Vで測定してあります。

### ミニマムモード(2)



マキシマムモード(1)



注記1： RDY信号は  $T_2$ 、 $T_3$ 、 $T_W$ ステートの終りでサンプルされ、必要に応じて  $T_W$ ステートを挿入します。

注記2： 最初と2番目の INTAサイクル間でカスケード・アドレスは確定します。

注記3： 2つの INTA サイクルがあり、2番目のINTAサイクルでは、MSM80C88A-10のローカル・アドレス/データバスは、フローティング状態となります。なお、ポインタ・アドレスに関する制御信号は、2番目のINTA サイクルについて示してあります。

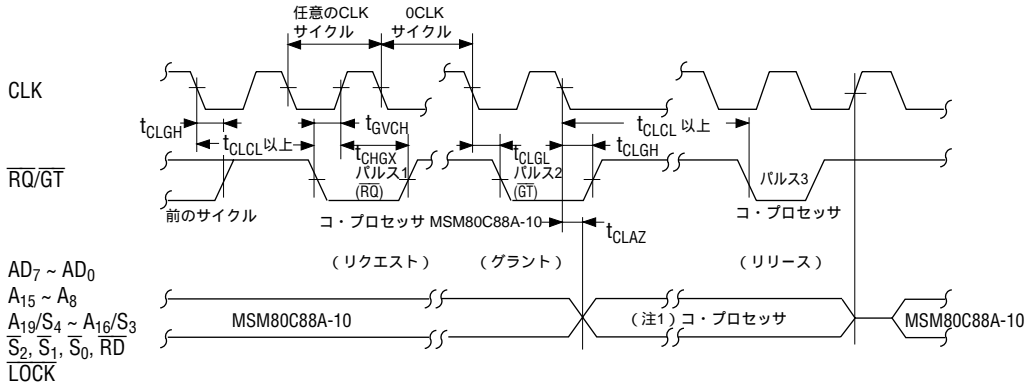
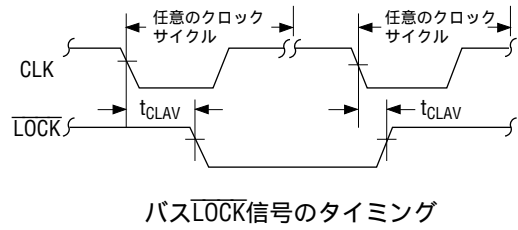
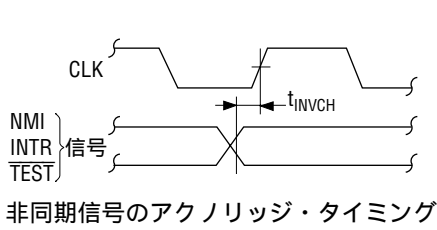
注記4： MSM82C84A、MSM82C88に関する信号は参考として示しました。

注記5： MSM82C88のコマンド制御信号 ( $\overline{MRDC}$ 、 $\overline{MWTC}$ 、 $\overline{AMWC}$ 、 $\overline{IORC}$ 、 $\overline{IOWC}$ 、 $\overline{INTA}$ 、DEN) の出力は、MSM82C88のCEN端子をアクティブ・ハイにした後、出力されます。

注記6： 特に示されている場合を除いて、すべてのタイミングは1.5Vで測定してあります。

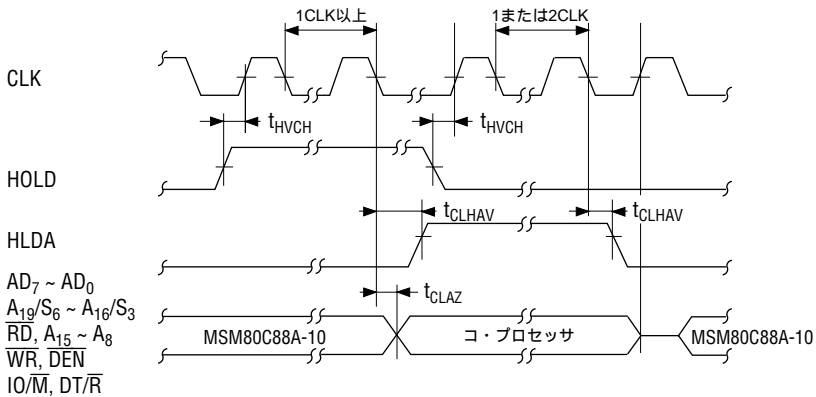
注記7：  $T_4$ サイクルより前にステータスはインアクティブになります。

#### マキシマムモード(2)

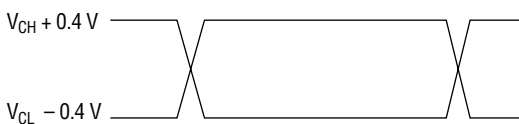


注記：1 コ・プロセッサは、この期間以外ではバスをドライブできません。（それ以外ではバスの衝突があります）

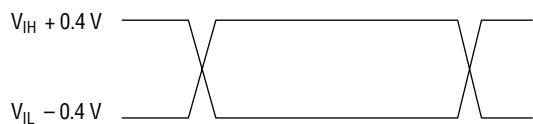
$\overline{\text{RQ}} / \overline{\text{GT}}$ のタイミング（マキシマム・モード時）



HOLD / HLDAのタイミング（ミニマム・モード時）



交流特性入力波形（CLK）



交流特性入力波形（CLK以外）

## ■ 端子説明

1. マキシマム・モード、ミニマム・モードの両モードで使用される端子

● AD<sub>7</sub> - AD<sub>0</sub> (双方向アドレス/データ・バス；入出力)

時分割のアドレス/データ・バスです。T<sub>1</sub>にはメモリ、あるいはI/O装置のアドレスを出力し、これ以外のときはデータ・バスとなります。割り込み受付応答時とホールド応答時には、高インピーダンス状態になります。

● A<sub>15</sub> - A<sub>8</sub>

アドレスの上位8ビットを出力します。この信号はバス・サイクル中変化しませんので、ALEでラッチする必要はありません。割り込み受付応答時とホールド応答時には、高インピーダンス状態になります。

● A<sub>19</sub>/S<sub>6</sub>、A<sub>18</sub>/S<sub>5</sub>、A<sub>17</sub>/S<sub>4</sub>、A<sub>16</sub>/S<sub>3</sub> (アドレス・バス/ステータス信号；出力)

T<sub>1</sub>ではメモリ・アドレスの上位4ビットを出力し(I/Oアクセス時はすべて“L”レベルを出力します)、これ以外のときはステータスを出力します。S<sub>3</sub>とS<sub>4</sub>は以下のようにどのセグメント・レジスタを使用したバス・サイクルかを符号化して示しています。

S <sub>4</sub>	S <sub>3</sub>	セグメント
0	0	エクストラ・セグメント
0	1	スタック・セグメント
1	0	コード・セグメント、未使用(I/O、割り込みベクタ)
1	1	データ・セグメント

S<sub>5</sub>は割り込み許可フラグの内容を示し、各クロックの立上りで更新されます。

S<sub>6</sub>は使用されておらず常に“L”レベルです。ホールド応答時には、高インピーダンス状態になります。

●  $\overline{RD}$  (読み出し指定信号；出力)

選択されたメモリ、あるいはI/Oからデータを読むときに“L”レベルになります。

(T<sub>2</sub>、T<sub>3</sub>、T<sub>W</sub>期間) また、T<sub>2</sub>ではAD<sub>7</sub> - AD<sub>0</sub>が高インピーダンス状態になるまで“H”レベルを保持しています。ホールド応答時には、高インピーダンス状態になります。

● READY (レディ信号；入力)

選択されたメモリ、あるいはI/Oがデータ転送を完了したことを示す信号です。

READYが“L”レベルだと、CPUはREADYが“H”レベルになるのを待ってバス・サイクルを終了します。READY入力は同期入力のためセット・アップ時間、及びホールド時間の両方を満足しなければなりません。

● INTR (マスク可能割り込み要求信号)

ソフト・ウェアでマスク可能な割り込み要求信号です。CPUは割り込み許可フラグが“1”の場合、各命令の最後のクロック毎にINTRをサンプリングしています。したがって、割り込みが受けられるまで“H”レベルを保持する必要があります。INTRはアクティブ“H”です。

●  $\overline{\text{TEST}}$  (動作同期信号; 入力)

CPUがWAIT命令を実行すると、CPUは $\overline{\text{TEST}}$ をサンプリングしはじめます。 $\overline{\text{TEST}}$ が“L”レベルならCPUは実行をそのまま続け、“H”レベルなら“L”レベルになるまで待機状態を続けます。待機状態では5クロック・サイクル毎に $\overline{\text{TEST}}$ 入力をサンプリングし、 $\overline{\text{TEST}}$ が“L”レベルであることを認識するとCPUはWAIT命令の次の命令から実行を再開します。

## ● NMI (マスク不可能割り込み要求信号; 入力)

ソフト・ウェアではマスク不可能な割り込み要求信号です。受けられると2型の割り込みを発生します。NMIはCPU内部で同期がとられ、立ち上りエッジが検出されるため、INTRのように受け付けを待つ必要はありません。ただし、非同期にNMI信号を発生させる場合は、2クロック・サイクル“H”レベルを保持する必要があります。

## ● RESET (リセット入力; 入力)

CPU内部の初期設定をする信号です。RESETはアクティブ“H”の信号で最低4クロックサイクル必要です。電源投入時は、電源電圧が最小電源電圧に達してから最低50 $\mu\text{s}$ の間“H”レベルを保持しなければなりません。

## ● CLK (クロック入力; 入力)

CPU内部の同期をとるための信号です。

電源投入時はリセットが完了するまでクロックを入力しなければなりません。

●  $V_{CC}$ 

電源端子です。フラット・パッケージでは $V_{CC}$ は3本ありますが、すべて接続することをお勧めします。しかし、2本しか接続できない場合は48番ピンと49番ピンに、またどうしても1本しか接続できない場合は48番ピンに接続して下さい。なお使用しない電源端子は必ず開放状態にして下さい。

## ● GND

接地端子です。GNDは2本ありますが、内部では接続されていないため、2本とも接地する必要があります。

● MN /  $\overline{\text{MX}}$  (モード設定端子; 入力)

ミニマム・モードで使用する場合は電源に接続します。

マキシマム・モードで使用する場合は接地します。

## 2. ミニマム・モードのみで使用される端子

● I /  $\overline{\text{MO}}$  (データ転送制御信号; 出力)

書込み、読み出しがメモリ空間に対してか、I/O空間に対してかを示します。“H”レベルではI/O空間を、“L”レベルではメモリ空間を示します。マキシマム・モード時の $S_2$ 、及びMSM80C86AのM /  $\overline{\text{IO}}$ とは極性が反転しています。ホールド応答時には、高インピーダンス状態になります。

●  $\overline{\text{WR}}$  (書出し指定信号; 出力)

選択されたメモリ、あるいはI/Oにデータを書き込むときに“L”レベルになります。(T<sub>2</sub>、T<sub>3</sub>、T<sub>W</sub>期間)。ホールド応答時には、高インピーダンス状態になります。

- $\overline{\text{INTA}}$  (割り込み受け付け応答信号; 出力)  
割り込み受け付けサイクルにおいて、 $\overline{\text{RD}}$ に代って割り込みベクタを読み取るのに使用されます。
- ALE (アドレス・ラッチ・イネーブル信号; 出力)  
時分割されて出力されるアドレス情報をラッチするためのストロブ信号で $T_1$ に出力されます。  
ALEは高インピーダンス状態にはなりません。
- DT /  $\overline{\text{R}}$  (データ送受制御信号; 出力)  
データ・バス・トランシーバを使用するときに必要な信号で、データ転送方向を制御します。  
“H”レベルでトランシーバからシステム・バスへ、“L”レベルでシステム・バスからトランシーバへの転送を示します。ホールド応答時には、高インピーダンス状態になります。
- $\overline{\text{DEN}}$  (データ・イネーブル制御信号; 出力)  
データ・バス上のデータが有効であることを示します。ホールド応答時には、高インピーダンス状態になります。
- HOLD (ホールド要求信号; 入力)  
HLDA (ホールド応答信号; 出力)  
他のバス・マスタ (例えばDMAコントローラ) がバスの使用权を要求する場合、HOLDを“H”レベルにします。すると、CPUは現在行っているバス・サイクルが完了した時点でHLDAを出力し、バスの使用权を放棄してホールド状態に入ります。CPUはこれ以降HOLDが“L”レベルになるまでバス・アクセスを停止しますが、バス・アクセスを必要とする (リード・サイクル、またはライト・サイクルが必要な場合、あるいは命令キューが空になった場合) までは実行を続けます。  
ホールド状態の間、CPUはローカル・バス<sup>注1</sup>と制御信号を高インピーダンス状態にします。また、HOLDは同期入力ですので、セット・アップ時間、ホールド時間を満足できない場合は外部に同期回路を必要とします。
- $\overline{\text{SS}}_0$  (ステータス信号; 出力)  
マキシマム・モード時の $S_0$ に相当する信号です。また、IO /  $\overline{\text{M}}$ 、DT /  $\overline{\text{R}}$ 、 $\overline{\text{SS}}_0$ をALEでラッチしデコードすることによってバスの制御信号が得られます。

IO / $\overline{\text{M}}$	DT / $\overline{\text{R}}$	$\overline{\text{SS}}_0$	バス・サイクルの内容
1	0	0	割り込み受け付け
1	0	1	I/Oからの読み出し
1	1	0	I/Oへの書き込み
1	1	1	ホルト
0	0	0	命令フェッチ
0	0	1	メモリからの読み出し
0	1	0	メモリへの書き込み
0	1	1	バス・サイクルを実行していません

### 3. マキシマム・モードのみで使用される端子

#### ● $\overline{S}_2$ 、 $\overline{S}_1$ 、 $\overline{S}_0$ (ステータス信号；出力)

MSM82C88に接続することでバスの制御信号が得られます。ステータスの内容は以下に示します。

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	バス・サイクルの内容
0	0	0	割り込み受け付け
0	0	1	I/Oからの読み出し
0	1	0	I/Oへの書き込み
0	1	1	ホルト
1	0	0	命令フェッチ
1	0	1	メモリからの読み出し
1	1	0	メモリへの書き込み
1	1	1	バス・サイクルを実行していません

ホールド応答時には、高インピーダンス状態になります。なお、この制御信号はリセット時に一度“H”レベルになった後、高インピーダンス状態になります。

#### ● $\overline{RQ}/\overline{GT}_1$ 、 $\overline{RQ}/\overline{GT}_0$ (バス要求制御信号；入出力)

ミニマム・モードのHOLD、HLDAに相当する信号ですが、ローカル・バス注1の使用権の授受は1本の信号線でハンド・シェイクが行われます。 $\overline{RQ}/\overline{GT}_0$ が $\overline{RQ}/\overline{GT}_1$ より優先度が高いことを除いて全く同一の機能です。

$\overline{RQ}/\overline{GT}$ によるバスの受渡し手順を以下に示します。

- (1) バス使用権を要求するバス・マスタは $\overline{RQ}/\overline{GT}$ に1クロック幅の“L”レベルを出力し、CPUからの許可を待ちます。
- (2) CPUはホールド状態の準備ができると、ローカル・バス注1とバス制御端子を高インピーダンス状態としホールド状態になります。これと同時にCPUは $\overline{RQ}/\overline{GT}$ に1クロック幅の“L”レベルを出力し、バス要求を出したバス・マスタに対しバス使用の許可を与えます。
- (3) バスを獲得したバス・マスタはバスの使用が終了すると $\overline{RQ}/\overline{GT}$ に1クロック幅“L”レベルを出力し、バス使用権の放棄をCPUに対して知らせます。CPUはこれを受けて、再びバスの使用権を獲得しバスの受渡しは完了します。

#### ● $\overline{LOCK}$ (バス・ロック信号；出力)

CPUは $\overline{LOCK}$ プレフィックス命令を実行すると $\overline{LOCK}$ 端子を“L”レベルにし、次の命令が終了すると、“H”レベルに戻します。バス・ロック信号を使用することにより、システムの共有データに対する排他制御を実現することができます。ホールド応答時には、高インピーダンス状態になります。

#### ● $QS_1$ 、 $QS_0$ (キュー・ステータス；出力)

命令キューの動作状態を示す信号です。これらの信号をデコードすることにより、CPUが現在いつフェッチした命令を実行中かを知ることができます。

内容を以下に示します。

$QS_1$	$QS_0$	命令キューの動作
0	0	キューの内容には変化がありません。
0	1	命令の第1バイトをキューから取り出しました。
1	0	キューの内容をすべてクリアしました。
1	1	命令の第2バイト以降をキューから取り出しました。

注記1： ローカルバスとは、MSM80C86Aに直接接続される時分割バスを意味します。

- HIGH (出力)

マキシマム・モード時は、常に“H”レベルです。

- バス・ホールド回路

バス・ホールド回路の採用で、アドレス/データバス、コントロール・バスにプルアップ抵抗をつける必要がなくなります。

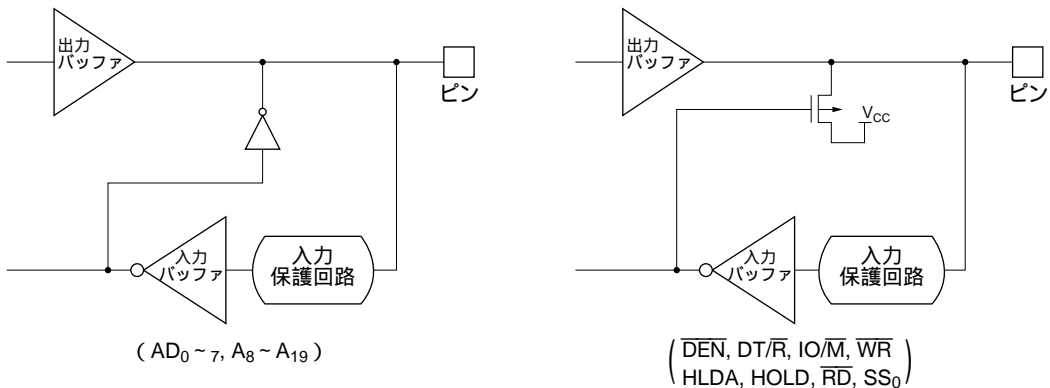
CMOS LSIにおいては、入力端子が高インピーダンスになると入力バッファに大電流が流れます。そのため、高インピーダンスになる可能性があるピンにはプルアップ抵抗をつける必要がありました。

ところが、プルアップ抵抗があるとその端子に“L”レベルが印加されているときは常に電流が流れるため、CMOSの低消費電力のメリットを十分に生かせませんでした。

しかし、バス・ホールド回路を採用することにより、このプルアップ抵抗が不要になり、プルアップ抵抗に流れる電流を無くすることができます。

バス・ホールド回路は、その端子に最後に印加された論理レベルを記憶し、入力端子が高インピーダンスにならないようにする機能があります。

また、その端子が高インピーダンスでない場合は、印加されている論理レベルを出力するため、プルアップ抵抗のように不必要な電流は流れません。バス・ホールド回路の付いた端子を反転させる場合は、最小 $I_{BHL0}$ または $I_{BHO}$ のドライブ能力が必要です。



バスホールド回路

## ■ 機能説明

### ● 汎用レジスタ

汎用レジスタとして16ビット・レジスタが8本用意されています。この汎用レジスタは4本ずつ2組のグループに分けられ、それぞれデータ・レジスタ、ポインタ/インデックス・レジスタと呼ばれています。

データ・レジスタは、それぞれ8ビット長×2、16ビット長のいずれでも使用でき、主にデータ操作の一時記憶に用いられます。

ポインタ/インデックス・レジスタは、16ビット長で使用されます。ポインタ/インデックス・レジスタの使用目的は、データ・レジスタよりはっきりしていますが、SPを除いてデータ・レジスタと同様に使用できます。（“アドレッシング・モード”の項を参照してください。）

### ● セグメント・レジスタ

MSM80C86A / MSM80C88A には1メガ・バイトのメモリ空間がありますが、これを最大 64 キロ・バイトのセグメントと呼ばれる窓を通して管理します。CPUは同時に 4 つのセグメントに直接アクセスすることができ、それらの先頭番地（ベース値）を示すのがセグメント・レジスタです。

CS（コード・セグメント）： 命令は常にこのセグメントからフェッチされます。

DS（データ・セグメント）： 主にプログラムのデータ領域に使用されます。

SS（スタック・セグメント）： スタック領域に使用されます。

ES（エクストラ・セグメント）： 特に使用目的は限定されていませんが、補助のデータ領域に使用すると便利です。

そのメモリ・サイクルでどのセグメントが参照されるかは暗黙の指定があります。

表2-1 セグメントの指定法

メモリ参照のタイプ	既定のセグメント	セグメント・オーバーライトによる変更	オフセット
命令フェッチ	CS	不可能	IP
スタック操作	SS	不可能	SP
データ・アクセス（以下の場合を除く）	DS	CS, ES, SS	実効アドレス
ストリング命令のソース・データ	DS	CS, ES, SS	SI
ストリング命令のデスティネーション・データ	ES	不可能	DI
BPをベース・レジスタとして使用した場合	SS	CS, DS, ES	実効アドレス

### ● 命令ポインタ

16ビット長の命令ポインタ（IP）は、MSM80C85Aにおけるプログラム・カウンタ（PC）に相当します。IPは現在実行中のコード・セグメントの先頭からのオフセット値（バイト単位の距離）を表し、次にフェッチすべき命令の論理アドレスを指しています。

## ●フラグ

フラグはすべて1ビットで構成され、算術演算あるいは論理演算の結果により変化する“ステータス・フラグ”6種類と、CPUの動作を制御する“コントロール・フラグ”3種類に分けられます。これら9種類のフラグは16ビットのフラグ・レジスタに格納されており、読み出したり、書き込んだりができるようになっています。この内、下位8ビットはMSM80C85Aとの互換性をとるために同じ順序で並んでいます。

## ステータス・フラグ

## CF：キャリー・フラグ

演算結果（8ビットまたは、16ビット）の最上位ビットに桁上げ、または桁借りが生じるとセットされます。

## ZF：ゼロ・フラグ

演算結果が零であるとセットされます。

## PF：パリティ・フラグ

演算結果の下位8ビットに“1”の数が偶数個あるとセットされます。主に、データ転送時に使用します。

## SF：符号フラグ

演算結果が2の補数表現で負数（最上位ビットが“1”）のときにセットされます。

## AF：補助キャリー・フラグ

演算結果（8ビット）の下位ニブルから上位ニブルへの桁上げ、または上位から下位の桁借りがあったことを示します。このフラグは10進算術演算命令時にCPU内部で使用されます。

## OF：オーバフロー・フラグ

演算結果の絶対値が大きすぎて桁あふれが生じるとセットされます。OFは有符号算術演算の桁あふれを示していますので、無符号算術演算のときは特に意味を持ちません。

有符号算術演算のときの桁あふれは、一般的にその場以外では補正不可能な場合ですから、INTO 命令で割り込みをかけ例外処理をすることができます。

## コントロール・フラグ

## DF：方向フラグ

DF をセットするとストリング操作命令はオート・デクリメント・モードで実行され、ストリングは高位アドレスから低位アドレスに向かって（インデックス・レジスタの値は減って行く）処理されます。クリアすると反対にオート・インクリメント・モードとなり、低位から高位に向かって処理がなされます。

## IF：割り込み許可フラグ

IF をセットすることでCPUはマスク可能な外部割り込み（INTR）の要求を受けられる状態になります。なお、内部割り込み、及びマスク不能割り込み（NMI）の制御はできません。また、S<sub>5</sub>（ステータス）でIFの内容を知ることができます。

## TF：トラップ・フラグ

TF はデバッグ時にCPUをシングル・ステップ動作させるためのフラグです。TFをセットすることにより、CPUは一命令実行毎に割り込みを発生します。これにより、CPU内部のレジスタを表示するルーチンなどを利用してプログラムの実行過程を細かくチェックすることができます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	OF	DF	IF	TF	SF	ZF	-	AF	-	PF	-	CF

図2-1 フラグ・レジスタの内容

## ● 割り込み

MSM80C86A / MSM80C88Aには外部割り込みと内部割り込みの2種類の割り込み機能があります。さらに外部割り込みにはマスク不能割り込み（NMI）とマスク可能割り込み（INTR）の2種類、また内部割り込みにはソフト・ウェア割り込みとハード・ウェア割り込みの2種類がそれぞれあります。それぞれの割り込みの優先度を表2-2に示します。

## 外部割り込み

〔INTR〕：マスク可能割り込みは、要求受け付け時にタイプ・ベクトルと呼ばれる割り込み原因を識別する8ビットのデータをCPUが読み込むことによりINTR 1本のピンで256種類の割り込みプロシジャーを用意することができます。

CPUはタイプ・ベクトルを受け取ると、フラグ、CS、IPをこの順でスタックにPUSHし、割り込みポインター・テーブルから割り込みプロシジャーの開始番地を読み込み、割り込みプロシジャーを開始します。

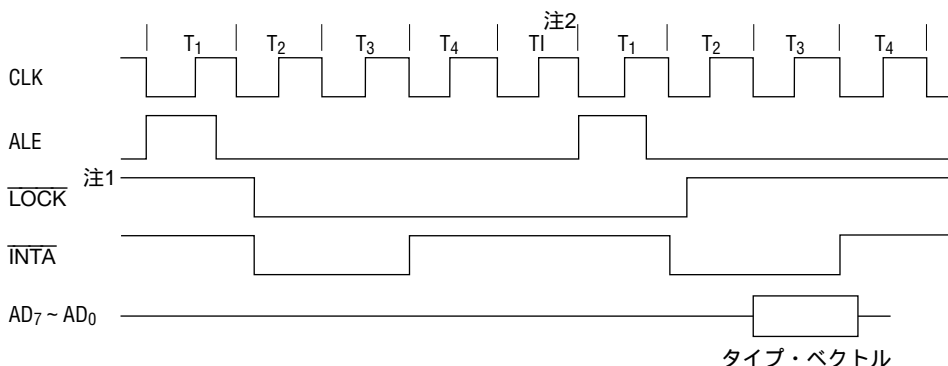
通常システムの割り込みには、MSM82C59A-2（プログラマブル割り込みコントローラ）を使用することで高機能で多様な割り込み処理を簡単に実現できます。

〔NMI〕：マスク不能割り込みは、IF フラグにより受け付けを禁止できません。主に電源異常、メモリ・エラー、バスのパリティ・エラーなど緊急事態が発生したことをCPUに知らせるために用意された割り込みです。ですから、INTR と違い割り込み要求は各クロックの立ち上りエッジでトリガがかかるようになっています。確実にNMI をかけるには2クロック幅のパルスが必要です。（INTRはINTAとのハンド・シェイク方式です。）

また、タイプ・ベクトルの値は2（“割り込みの型が2型である”と呼びます。）に決められていますので、割り込み受け付けバス・サイクル（INTA サイクル）は実行されません。

表2-2 割り込み優先度

割り込みの種類		型	優先度	
外部割り込み	NMI	2	2	
	INTR	0 ~ 255	3	
内部割り込み	ソフト・ウェア割り込み	INTn	0 ~ 255	1
		INT0	4	1
		INT3	3	1
	ハード・ウェア割り込み	シングル・ステップ割り込み	1	4
		除算エラー	0	1



注1. マキシマム・モードのみ

注2. 数回のアイドル・クロックが入ることがあります。このとき、バスはドライブされます。

図2-2 割り込み受け付けサイクル

### 内部割り込み

内部割り込みには3種類のソフト・ウェア割り込みと、2種類のハード・ウェア割り込みがありますが、いずれの内部割り込みもIFフラグにより割り込みを禁止することはできません。また、割り込みシーケンスにINTAサイクルはありません。

〔ソフト・ウェア割り込み〕：

#### INTn 命令

2バイト命令です。1バイトのオペランドで0～255の割り込みの型を指定します。一種のスーパバイザ・コールとして使用すると便利です。

#### INTO命令

INTO命令を実行すると、CPUはOFフラグをチェックし、セットされているときは4型の割り込みを発生します。これは有符号算術演算を実行したときのオーバーフロー・エラー処理のときに使用します。

#### INT3命令

1バイト命令で3型の割り込みを発生します。この命令は、MSM80C85Aの“RST7”命令と同様にブレーク・ポイントの設定に使用すると便利です。

〔ハード・ウェア割り込み〕：

#### シングル・ステップ(トラップ)割り込み

TFフラグをセットすることにより、CPUは1命令実行する毎に1型の割り込みを発生します。例えば、1型の割り込みルーチンとして、レジスタや重要なメモリの内容を表示するプログラムを用意しておけば、プログラマはプログラムの過程を細かく知ることができ、より容易にデバッグをすることができます。その上、INT3命令を使ってブレーク・ポイントを設定しておけば、より複雑なデバッグ処理ができます。

TFフラグをセットしたり、クリアしたりする命令はありませんが、PUSHF命令、POPF命令を使うことによりTFフラグを操作することができます。また、割り込みによってTFフラグはクリアされますので、シングル・ステップ用ルーチン中はシングル・ステップ動作をしません。

#### 除算エラー割り込み

除算命令(DIVまたはIDIV)を実行した結果、商が指定したデスティネーションより大きくなった場合は直ちに0型の割り込みが発生します。除算エラー割り込みを禁止することはできませんから、エラー検出の必要の有無にかかわらず処理ルーチンを用意する必要があります。

### ● 割り込み受付サイクル

MSM82C59A-2(プログラマブル割り込みコントローラ)を使用した場合の割り込み受け付けサイクルについて説明します。

CPUはMSM82C59A-2からの割り込み要求を受けけると、 $\overline{\text{INTA}}$ 端子に2回の“L”レベルを出力します。最初の“L”レベルで、MSM82C59A-2に割り込みを受け付けたことを知らせ、これを受けたMSM82C59A-2は、その割り込みに対応するタイプ・ベクトルをデータ・バスに出力します。2回目の割り込み受け付け信号の立ち上りでCPUはタイプ・ベクトルを読み込みます。これに引き続きCPUは、フラグ、CS、IPの順でレジスタをスタックに退避し、IFフラグをクリアします。

次に、割り込みポインタ・テーブルからタイプ・ベクトルに対応した新しいIP、CSを読み込み、割り込み処理ルーチンへ実行を移します。また、割り込み処理ルーチンから戻る時は、必ずIRET命令を使用しなければなりません。

それは、割り込み処理ルーチンをコールする場合は、CS、IPの他にフラグモスタックに退避するからです。

割り込み受け付けサイクルを図2-2 に示します。

#### ●物理アドレスと論理アドレス

MSM80C86A / MSM80C88Aには物理アドレスと論理アドレスの2つのアドレスが存在します。物理アドレスとは、1メガ・バイトのメモリ空間の中に実際に割り付けられているメモリの番地で、0番地から FFFFFFFH 番地まで存在します。

論理アドレスは、全メモリ空間の中に開けた64キロ・バイトの“窓”の中での相対的なアドレスを示しています。この“窓”をセグメントと呼び、CPUはこの中でしかアクセスできませんが、セグメントは全メモリ空間を16バイト毎にどこでも移動させることができますから、CPUは全メモリ空間を自由にアクセスできることとなります。

このセグメントは同時に4つ定義でき、それぞれの位置は全く独立に設定できます。したがって、一部、あるいはすべてが重なりあっても良く、同一のアドレスを複数のセグメントが共有することもあります。

セグメントという概念を用いることにより、プログラマは特に意識することなくプログラムのモジュール化ができ、非常に大きなプログラムを一体構造で書くことも少なくなるでしょう。物理アドレスと論理アドレスとの間には次の関係があります。

$$\text{物理アドレス} = \text{セグメント} \cdot \text{ベース値} + \text{論理アドレス}$$

ここで、セグメント・ベース値とは、セグメントの先頭の物理アドレスを意味し、セグメント・レジスタの値を16倍した値です。つまり、論理アドレスとはセグメントの先頭からの距離（バイト単位）ということになります。この変換はCPUが自動的に行ってくれます。

#### ●スタック

MSM80C86A/MSM80C88A のスタックはメモリ中に置かれ、スタック・セグメント・レジスタ（SS）とスタック・ポインタ・レジスタ（SP）により位置が示されます。1つのシステムが持つスタックの数に制限はありませんが、同時にいくつものスタックをアクセスすることはできません。また、1つのスタックの最大長は64キロ・バイトまでです。

ここで注意が必要なことは、スタックのセグメント・ベース値がそのスタックの“底”ではないということです。したがって、セグメント・ベース値が実際にスタックのある場所と離れていることもあります。

また、MSM80C86Aの場合は、スタック・ポインタ・レジスタの初期値は偶数に設定しておく方がCPUの実行速度の点からいって有利です。

（“MSM80C86A と MSM80C88A のメモリ・アクセスの違い”の項を参照してください。）

#### ●MSM80C86AとMSM80C88Aのメモリ・アクセスの違い

MSM80C88Aは、外部に対しては8ビットでインターフェイスをとっています。そのため、メモリの配列は従来の8ビットCPU（MSM80C85Aなど）と全く同じで、またメモリへのアクセスの方法も順番に1バイトずつ行います。

しかし、MSM80C86Aは外部に対しても16ビットでインターフェイスをとっていますので、MSM80C88Aとはメモリの配列が異なります。MSM80C86Aは、偶数バンクと奇数バンクの2つのメモリ群（512キロ・バイト×2）の2次元配列になっています。したがって、偶数アドレスから始まるワード・データは1回のパス・サイクルでアクセスできますが、奇数から始まる場合は2回必要になります。（図2-3）

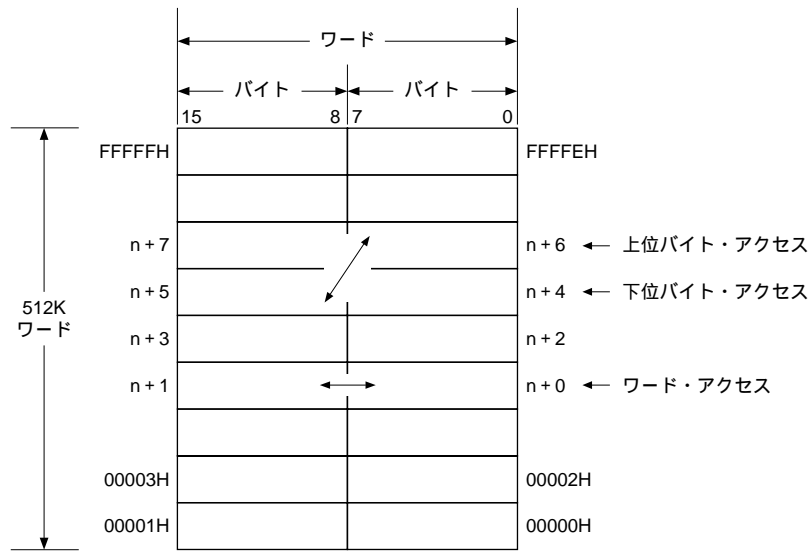


図2-3 MSM80C86Aのメモリ・アクセス

これは、ワード・データは偶数アドレスから格納した方が良いことを示しています。特に、スタック操作、及びブロック転送ではこの影響が大きくて、たった1バイト節約したためにシステムの実行速度が落ちてしまいます。しかし、実行速度が低下する以外はすべて同じで、プログラム自体には全く影響を与えません。

また、MSM80C88Aの場合もMSM80C86Aと互換性を考えれば、ワードデータは偶数アドレスから格納した方がよいでしょう。

### ●メモリ構成

MSM80C86A / MSM80C88Aは1メガ・バイトのメモリ空間と64キロ・バイトのI/O空間を持っていますが、その一部の使用法は決められています。(表2-3)

システム・リセットをかけると、CPUはFFFF0H番地から実行を開始します。そこで、一般的にはメモリ空間の最高位アドレス部にはROMを割り当てます。また、割り込みポインタ・テーブルがメモリ空間の最低位(0H~7FH)に位置します。割り込みポインタ・テーブルはプログラム毎に違ったり、同一のプログラムでも実行中に書き変えることがあります。そこで、一般的にはメモリ空間の最低位アドレス部にはRAMを割り当てます。

表2-3 予約済のメモリアドレス

[メモリ]:

0H ~ 13H (20バイト): 内部割り込み用ポインタ・テーブル

### ●モードの選択

MSM80C86A / MSM80C88Aにはミニマム・モード、及びマキシマム・モードと呼ばれる2つのモードがあります。動作モードを変えることにより、CPUはいろいろな規模のシステムに対して柔軟に対応することができます。

[ミニマム・モード] ( $MN / \overline{MX} = "H"$ )

すべてのバス制御信号はCPUから出力されます。このモードは、小さなシングル・プロセッサ・システム（およそ基板1~2枚）に向いています。

[マキシマム・モード] ( $MN / \overline{MX} = "L"$ )

このモードは大きなシステム、特にマルチ・プロセッサ・システム等に向いています。バス制御信号は3ビットにコード化されて出力されます。この信号はMSM82C88（バス・コントローラ）によりデコードされ、拡張バス制御信号となりシステムに供給されます。これにより、空いた端子をシステム内のほかのCPUとの協調動作制御に使用することができます。

### ●リセット

リセット入力によりCPUの内部は表2-4のようになります。したがって、CPUはFFFF0H番地から命令を実行し始めます。通常は、ここにセグメント間直接ジャンプ命令を置いて、本来のシステム・プログラムのスタート番地へ飛ばせます。

また、IFフラグはクリアされていますから、マスク可能割り込み（INTR）はIFフラグがセットされるまで受け付けられません。

リセット・パルス幅は、最低4クロック幅必要です。さらに、電源投入時はCLKが安定して入力し、電源電圧が動作範囲に立ち上がるまでアクティブにする必要があります。

リセット動作中はNMI、 $\overline{RQ} / \overline{GT}$ 、HOLDの要求はいずれも受け付けられませんが、リセット直後の $\overline{RQ} / \overline{GT}$ 、HOLDの要求は最初の命令フェッチが行われる前に受け付けられます。

表2-4 リセット後のCPUの状態

内部レジスタ	内容
フラグ	クリア
IP	0000H
CS	FFFFH
DS	0000H
SS	0000H
ES	0000H
命令キュー	空

表2-5 リセット中の出力端子の状態

端子名	状態	端子名	状態
AD <sub>15</sub> -AD <sub>0</sub>	高インピーダンス状態	AD <sub>7</sub> -AD <sub>0</sub>	高インピーダンス状態
A <sub>19</sub> /S <sub>6</sub> -A <sub>16</sub> /S <sub>3</sub>	高インピーダンス状態	A <sub>19</sub> /S <sub>6</sub> -A <sub>16</sub> /S <sub>3</sub> , A <sub>15</sub> -A <sub>8</sub>	高インピーダンス状態
BHE/S <sub>7</sub>	高インピーダンス状態	$\overline{SS}_0$	高インピーダンス状態
$\overline{S}_2$ - $\overline{S}_0$	"H" 出力後高インピーダンス状態	$\overline{S}_2$ - $\overline{S}_0$	"H" 出力後高インピーダンス状態
M/ $\overline{IO}$ , DT/ $\overline{R}$ , DEN	"H" 出力後高インピーダンス状態	IO/ $\overline{M}$ , DT/ $\overline{R}$ , DEN	"H" 出力後高インピーダンス状態
LOCK, WR, RD	"H" 出力後高インピーダンス状態	LOCK, WR, RD	"H" 出力後高インピーダンス状態
$\overline{INTA}$	"H"	$\overline{INTA}$	"H"
ALE, HLDA	"L"	ALE, HLDA	"L"
RQ, GT1, 0	"H" 出力後入力状態	RQ / GT1, 0	"H" 出力後入力状態
QS <sub>1</sub> , QS <sub>0</sub>	"L"	QS <sub>1</sub> , QS <sub>0</sub>	"L"

(a) MSM80C86A

(b) MSM80C88A

### ● HALT

HALT命令を実行するとCPUはホルト状態になります。ホルト状態のCPUは、 $T_1$ 状態におかれていますので、出力を高インピーダンス状態にすることはありませんが、HOLD、あるいは $\overline{RQ}$ / $\overline{GT}$ 要求は正常に受け付けます。ホルト状態を解除するにはリセット、あるいは外部割り込みが必要です。ホルト状態は、外部割り込み待ちの無限ループとして使用されます。

### ● READY

アクセス時間の大きいメモリ、あるいはI/O装置を使用した場合、READY入力を“L”レベルにすることでバス・サイクルを延長することができます。 $T_2$ 状態が終る前にREADY入力を“L”レベルにすると、CPUは $T_W$ 状態になり、メモリ、あるいはI/O装置が動作を終了するのを待ちます。(Tw状態とは、メモリ、あるいはI/O装置の動作終了を待っている状態のことで、 $T_3$ 状態を繰り返しています。)  $T_3$ 状態を終了するには、CLKの立ち上りエッジより前にREADY入力を“H”レベルにします。

また、 $T_3$ 状態でCLKが“L”レベルの間は、READY入力を“H”レベルから“L”レベルに変えることはできません。

READY入力の与えかたには、次の2通りの方法があります。

#### 〔通常ノット・レディーのシステム〕

メモリ、あるいはI/O装置がCPUから制御信号( $\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{INTA}$ )を受け取ると、これに応答する準備が完了した時点で、周辺素子はREADY入力に対して“H”レベルを送り返します。これによりCPUはバス・サイクルを終了します。

この方法は、各周辺素子にREADY信号の発生回路が必要になりますが、仮にREADY信号が遅れても $T_W$ 状態が1回挿入されるだけで誤動作にはつながりません。

#### 〔通常レディーのシステム〕

メモリ、あるいはI/O装置は $T_W$ 状態なしで動作できると仮定し、 $T_W$ 状態が必要な場合は、選択された時点でREADY入力を“L”レベルにし $T_W$ 状態の挿入が必要であることをCPUに知らせます。この方法は、 $T_W$ 状態を必要としない周辺素子はREADY信号の発生回路がいらないので、論理回路を減らすことができます。しかし、 $T_W$ 状態を必要とする周辺素子がREADY入力を“L”レベルにするのが遅れると、データの受渡し完了しないうちにバス・サイクルが終了してしまいます。したがって、この方法を採用する場合は、システムのタイミングを十分に検討しておく必要があります。

### ● バス・ロック

MSM80C86A / MSM80C88Aはマキシマム・モードのとき、バス・ロック信号を使用することによりシステムの共有データに対する排他制御を行うことができます。

CPUは1バイトのLOCKプレフィックスを実行すると、それに続く命令が終了するまでバス・ロック信号を“L”レベルにします。この間CPUに、バスの使用要求( $\overline{RQ}$ / $\overline{GT}$ 要求)があっても記憶しておくだけで要求の受け付けは待たされます。また、共有バス上のバス調停回路にバス・ロック信号を与えることにより、システム・バスの排他制御を行うことができます。ミニマム・モードのとき、CPUはバス・ロック信号を出力しませんが、LOCKプレフィックスを使用するとHOLD要求に対するHLDAの応答を待たすことができます。

### ● WAIT命令と $\overline{\text{TEST}}$

MSM80C86A / MSM80C88Aは、WAIT命令と動作同期信号 ( $\overline{\text{TEST}}$ ) を用いることによって外部事象に動作を同期させることができます。この機能は8087 (数値データ・プロセッサ) の演算終了待ちに使用すると便利です。

### ● ESC (エスケープ)

ESC命令は、MSM80C86A / MSM80C88Aに対してスレーブ状態のプロセッサにデータ、あるいは命令を渡すための命令で、6ビットのオペランド・フィールドを使用してスレーブ・プロセッサの命令を自由に定義できます。この命令を使うと、プログラム上スレーブ・プロセッサをあたかもMSM80C86A / MSM80C88Aであるかのように扱えます。現在この命令は、8087 (数値データ・プロセッサ) で応用されています。ここで注意が必要なことは、MSM80C86A / MSM80C88AはメモリからESC命令をフェッチした時点と、実行される時点が一致しないということです。しかし、これはMSM80C86A / MSM80C88Aのキュー・ステータス信号 ( $QS_1, QS_0$ ) を監視することで解決できます。

### ● アドレッシング・モード

MSM80C86A / MSM80C88Aには高級言語のデータ構造を効率よく表現できるように、6タイプ24種類アドレッシング・モードが備えられています。それぞれのアドレッシング・モードにより計算されたアドレスを実効アドレスと呼びます。

それぞれのアドレス計算に必要なクロック数を表2-6に示します。

#### 〔レジスタ・モード〕

データ操作をレジスタ内に限るモードです。このモードを使用すると、命令コードを短くでき、実行速度も速くできます。

#### 〔直接モード〕

固定されたアドレスを直接指定するモードです。主に、単純変数をアクセスするのに使用します。

#### 〔レジスタ間接モード〕

このモードは、レジスタの内容がアドレスを表しているアドレッシングです。このモードを使うと、レジスタの内容を変えることにより様々なアドレスにアクセスすることができます。特に、このレジスタの内容をポインタ変数と呼ぶことがあります。

#### 〔ベース・モード〕

ベース・レジスタの内容とディスプレイメントの値の和を実効番地にするモードです。このモードは構造体を表現したり、スタック内の引数を扱うのに有効です。構造体にアクセスするには、BXレジスタに目的の構造体の先頭番地を格納し、ディスプレイメントに必要な項目のオフセットを表します。このようにすれば、BXレジスタの値を変えるだけで他の要素にアクセスすることができます。またスタック内の引数は一種の構造体ですから、BXレジスタの代わりにBPレジスタを使用すれば容易にスタック内を操作できます。(BPレジスタを選択するとスタック・セグメントが選択されます。)

#### 〔インデックス・モード〕

インデックス・レジスタの内容とディスプレイメントの値の和を実効番地にするモードです。このモードは配列の要素にアクセスするのに有効です。ディスプレイメントで配列の先頭番地を示

し、そこから目的の要素までのオフセットをインデックス・レジスタで表します。配列の各要素の大きさは同じですから、インデックス・レジスタには「要素番号(0から始まる)×1要素のバイト数」が入ります。

#### 〔ベース・インデックス・モード〕

ベース・レジスタ+インデックス・レジスタ+ディスプレイメントを実効番地にするモードです。このモードは、配列を含む構造体にアクセスするのに有効です。

また、2つのポインタ変数を持ちますから、これ以外にも柔軟な応用が考えられます。

この他にも2種類のアドレッシングの方法があります。

#### 〔ストリング・モード〕

ストリング命令で使用されるのがストリング・モードです。このモードは他のモードと異なり、指定しなくてもインデックス・レジスタが選択されます。SIレジスタにソース・ストリングの先頭番地を、またDIレジスタにデスティネーション・ストリングの先頭番地をそれぞれセットしてからストリング命令を実行します。反復操作のときは、CXレジスタで示されるバイト、あるいはワード数だけSI、DIレジスタをそれぞれ順次変化させながら実行を繰り返します。

#### 〔I/Oモード〕

I/O空間に割り付けられたポートを指定するには2通りの方法があります。1つは、命令の中で1バイトのオペランドにより指定する直接モードで、固定されたポートにアクセスするのに用いられます。もう一つはDXレジスタの内容でポートを指定するレジスタ間接モードです。このモードを使用すると0～65,535までのポートが使用できます。

また、メモリ空間に割り付けられたポートを指定するときは、メモリに対するのと全く同じアドレッシング・モードが使用できます。

表2-6 アドレス計算時間

アドレス構成要素	使用レジスタ	クロック数(注1)
ディスプレイメントのみ	disp(注2)	6
ベース/インデックスのみ	BX, BP, SI, DI	5
ディスプレイメント + ベース/インデックス	BX, BP, SI, DI + disp	9
ベース + インデックス	BP + DI, BX + SI	7
	BP + SI, BX + DI	8
ディスプレイメント + ベース	BP + DI + disp BX + SI + disp	11
ディスプレイメント + ベース + インデックス	BP + SI + disp BX + DI + disp	12

注記1: セグメント・オーバーライド・プレフィックスがある場合は、2クロック余計に必要です。

注記2: dispは、ディスプレイメントを表します。

#### ●使用上の注意

1. インテル社8087-1(10MHz)とは直接インターフェースできません。

## ■ 付録

### ● 低速版から高速版への置き換え時の注意事項

従来生産しておりました低速版は、下記表のとおり高速版に集約されております。低速版をお使いの方で高速版に置き換えをされる方は、次ページ以降の置き換え時の注意事項をお読みください。

高速版製品名（集約製品）	低速版製品名（廃品）	備考
M80C85AH	M80C85A/M80C85A-2	8bit MPU
M80C86A-10	M80C86A/M80C86A-2	16bit MPU
M80C88A-10	M80C88A/M80C88A-2	8bit MPU
M82C84A-2	M82C84A/M82C84A-5	クロックジェネレータ
M81C55-5	M81C55	RAM,I/O,タイマ
M82C37B-5	M82C37A/M82C37A-5	DMAコントローラ
M82C51A-2	M82C51A	USART
M82C53-2	M82C53-5	タイマ
M82C55A-2	M82C55A-5	PPI

## MSM80C88A-10とMSM80C88A-2、MSM80C88Aとの相違点

### 1) 製造プロセス

全て使用プロセスは1.5 $\mu$ Si-CMOSで同じであり問題ありません。

### 2) 設計

回路のタイミングを若干変更してありますが、チップサイズは同一で論理は全く同じになっており問題ありません。

### 3) 電気的特性

96年度マイクロプロセッサデータブックより、MSM80C88A-10はMSM80C88A-2、MSM80C88Aの規格も満足することを明記しており問題ありません。

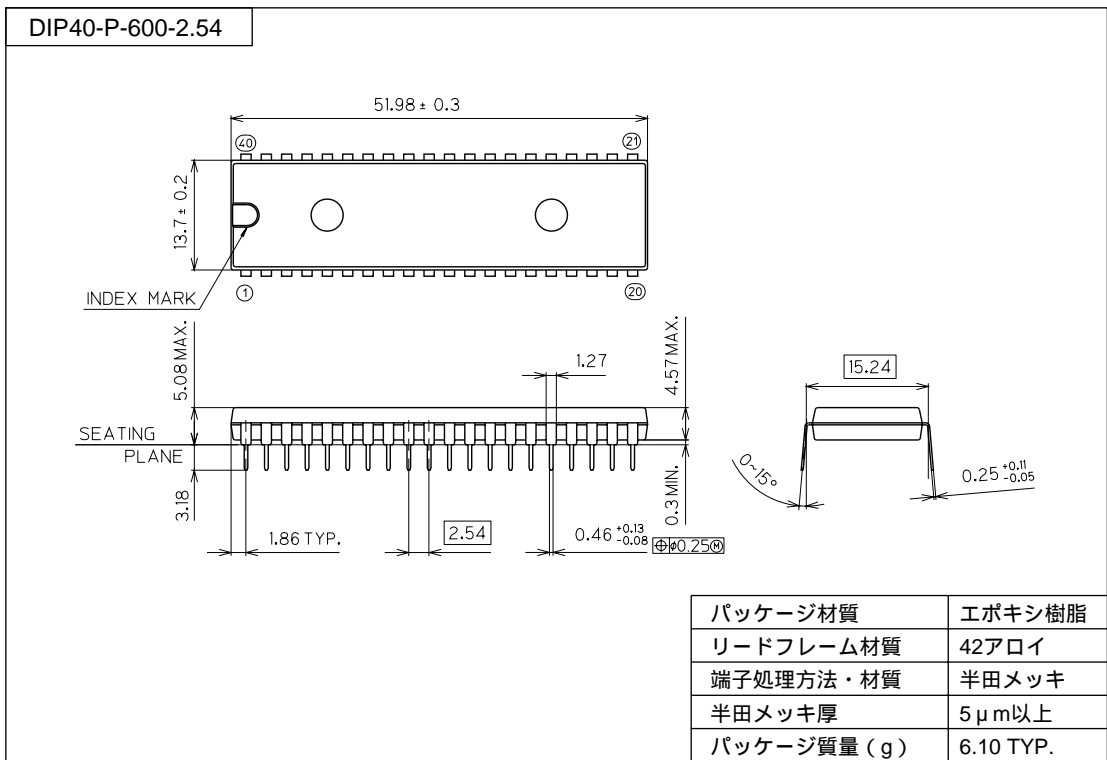
### 4) その他の注意事項

- (1) MSM80C88A-10は、10MHz用に高速化設計されているため、若干ノイズ特性が異なることも考えられますので、置き換え時にはノイズ関連の評価をすることを推奨します。
- (2) MSM80C88A-10は、MSM80C88A-2、MSM80C88Aのスペックを満足しておりますが、タイミング的に若干異なるため、クリティカルなタイミング設計をされている場合は、温度、電圧などを振って動作マージンの評価をされることを推奨します。

以上

## ■ パッケージ寸法図

(単位 : mm)

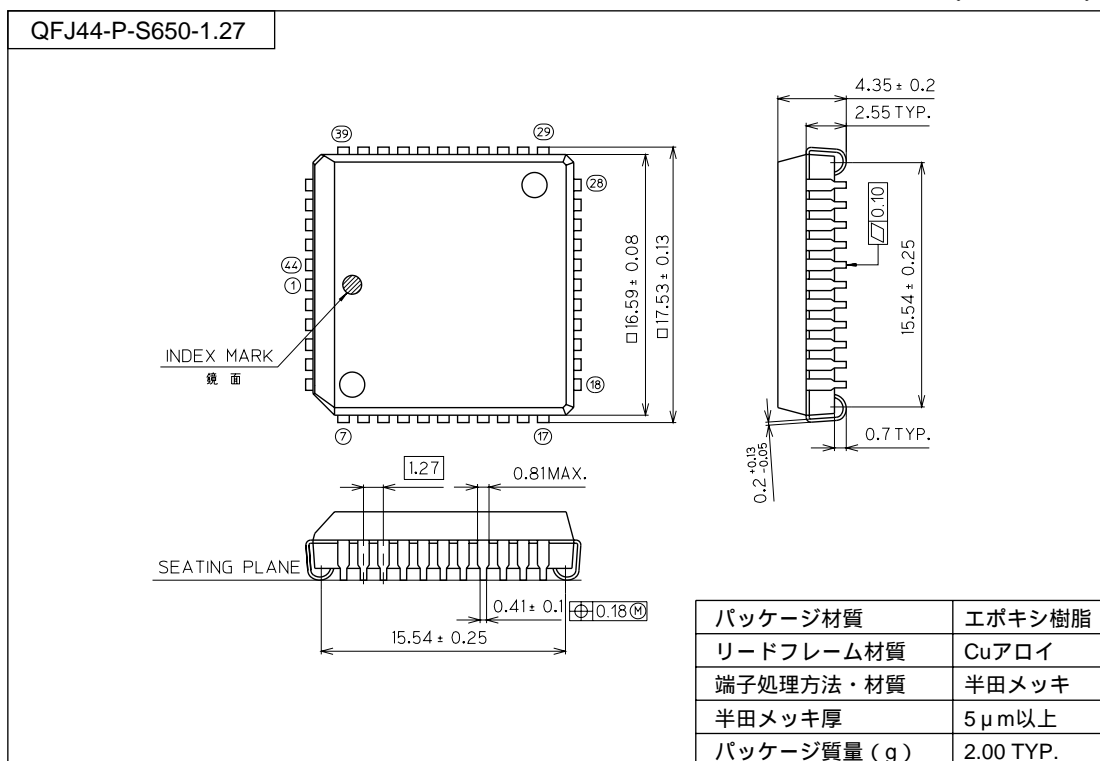


### 表面実装型パッケージ実装上のご注意

SOP、QFP、TSOP、TQFP、LQFP、SOJ、QFJ (PLCC)、SHP、BGA等は表面実装型パッケージであり、リフロー実装時の熱や保管時のパッケージの吸湿量等に大変影響を受けやすいパッケージです。

したがって、リフロー実装の実施を検討される際には、その製品名、パッケージ名、ピン数、パッケージコード及び希望されている実装条件 (リフロー方法、温度、回数)、保管条件などを弊社担当営業まで必ずお問い合わせください。

(単位 : mm)

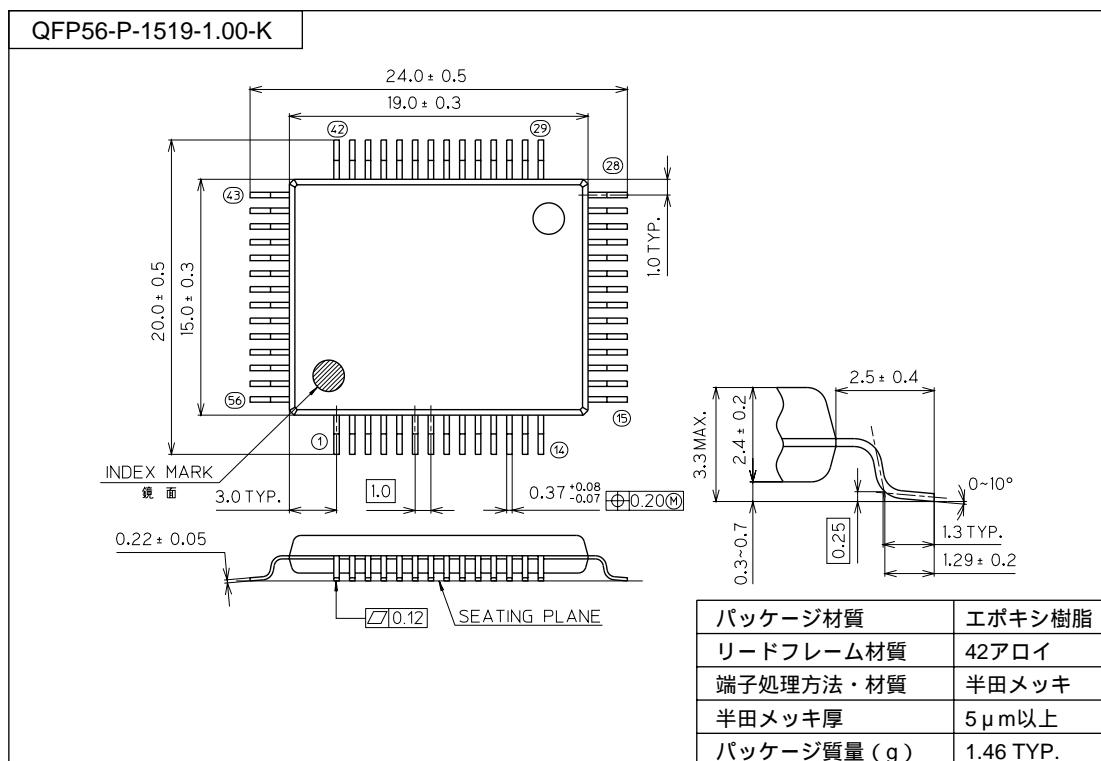


#### 表面実装型パッケージ実装上のご注意

SOP、QFP、TSOP、TQFP、LQFP、SOJ、QFJ (PLCC)、SHP、BGA等は表面実装型パッケージであり、リフロー実装時の熱や保管時のパッケージの吸湿量等に変化を受けやすいパッケージです。

したがって、リフロー実装の実施を検討される際には、その製品名、パッケージ名、ピン数、パッケージコード及び希望されている実装条件 (リフロー方法、温度、回数)、保管条件などを弊社担当営業まで必ずお問い合わせください。

(単位 : mm)



#### 表面実装型パッケージ実装上のご注意

SOP、QFP、TSOP、TQFP、LQFP、SOJ、QFJ (PLCC)、SHP、BGA等は表面実装型パッケージであり、リフロー実装時の熱や保管時のパッケージの吸湿量等に大変影響を受けやすいパッケージです。

したがって、リフロー実装の実施を検討される際には、その製品名、パッケージ名、ピン数、パッケージコード及び希望されている実装条件 (リフロー方法、温度、回数)、保管条件などを弊社担当営業まで必ずお問い合わせください。