SX18ACマイコンモジュールキット

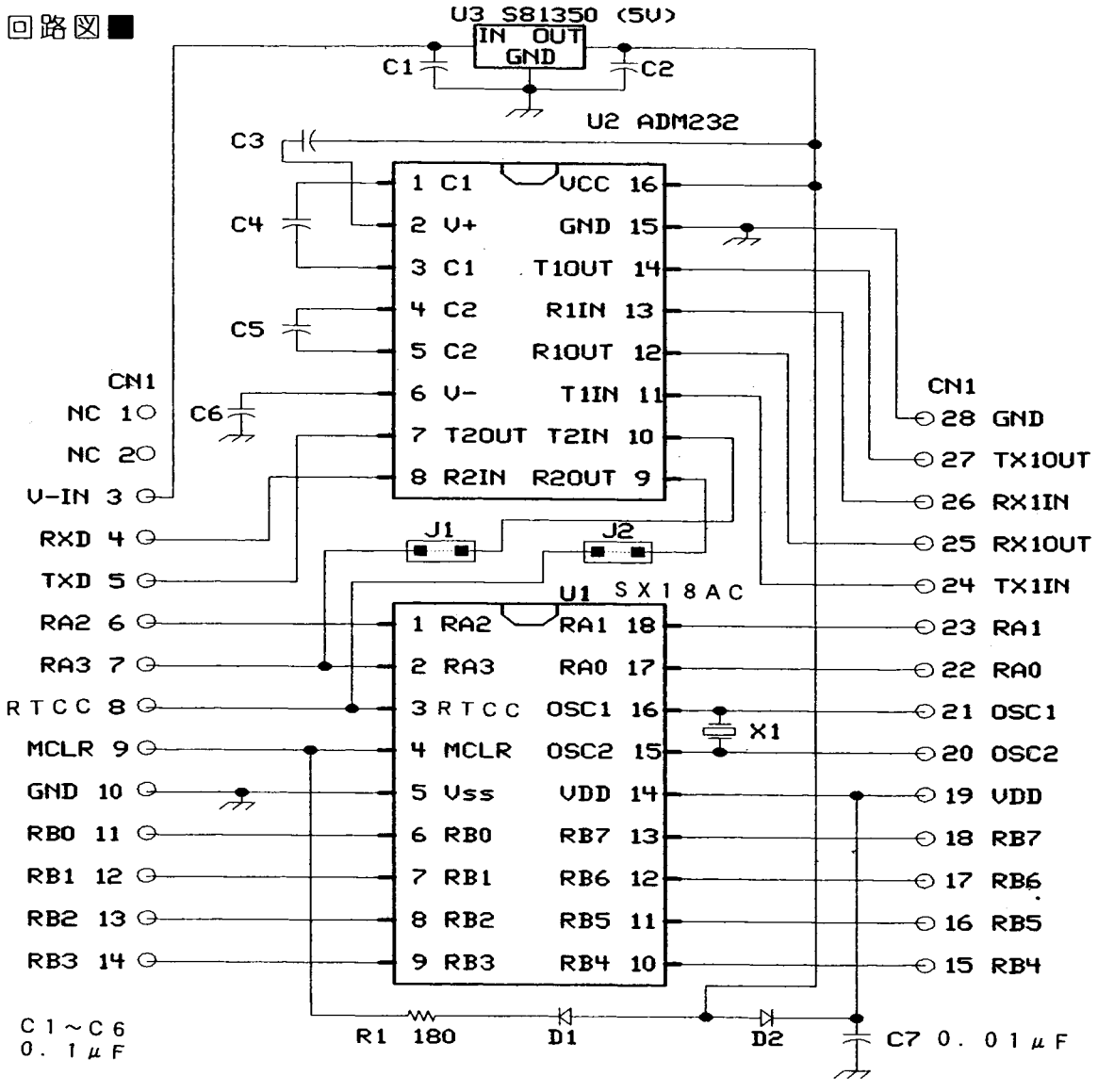
超高速50MHz動作マイコンSX18ACが
マイコンモジュールキットになりました。

■SX18ACはフラッシュROMですので、10,000回以上
　の書き替えができ、プログラム開発が容易にできます。
■プログラム書替えの為のピンがすべて出ていますので、
　この基板に実装したまま書き込みが出来ます。
　（専用のライターが必要です。）

ＳＸ１８ＡＣ超高速５０ＭＨｚ
マイコンモジュールキット　５×１．８ｃｍ
ＲＳ２３２搭載，５０ＭＨｚ動作

■ＰＩＣ１６Ｃ５４，Ｆ８４ピン互換の超高速５０ＭＨｚ動作マイコンＳＸ１８ＡＣがマイコンモジュールキットになりました。
■ＳＸ１８ＡＣはフラッシュＲＯＭですので、１００００回以上の書き替えができ、プログラム開発が容易にできます。
■プログラム書き込みの為のピンがすべて出ていますので、この基板に実装したまま書き込みが出来ます。（専用のライターが必要です。）
■ＲＳ２３２用ＩＣを搭載していますので、パソコン等とのデータ通信などがこの基板のみで行なえます。
■５Ｖレギュレータ１Ｃ，セラミック発振子が付いており、この基板に電源を接続するだけで動作します。

■回路図■



■部品表■

番号	名称	数	備考
Ｃ１－６	ｺﾝﾃﾞﾝｻ ０．１μＦ	６	積層セラミックコンデンサ　（１０４）
Ｃ７	ｺﾝﾃﾞﾝｻ 4700~10000pF	１	ｾﾗﾐｯｸｺﾝﾃﾞﾝｻ (472)~(103)
Ｄ１，２	ダイオード	２	ショットキーダイオード
Ｒ１	抵抗１８０Ω	１	１／６Ｗ　茶灰茶金
Ｕ１	ＳＸ１８AC/DP	１	５０MHz動作　ROM=2Kﾜｰﾄﾞ RAM=138ﾊﾞｲﾄ
Ｕ２	ＡＤＭ２３２	１	２３２レベルコンバータ
Ｕ３	Ｓ８１３５０	１	低ドロップ５Ｖレギュレータ
Ｘ１	セラロック	１	セラミック発振子５０ＭＨｚ（ｺﾝﾃﾞﾝｻ内蔵）
基板	AE-PIC18	１	両面スルーホール基板
	連結ソケット	１	２８ＰＩＮ用
	ピンヘッダ		Ｊ１、Ｊ２用

■製作■

あらかじめ部品表と部品をてらしあわせ、数等をチェックしてから製作してください。連結ソケット以外の部品は部品面側（白い印刷のある面）にとりつけます。

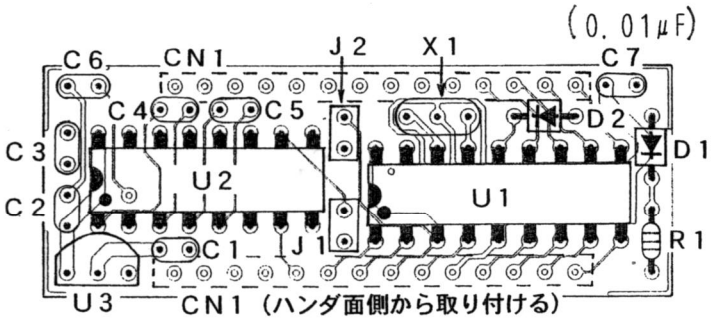
①コンデンサ、抵抗、ダイオードの順で半田付けしていきます。
　コンデンサの足が５ｍｍピッチになっている場合はあらかじめ、２．５ｍｍピッチになるようにピンセット等で加工してください。
　ダイオードには極性がありますので基板印刷のマークにあわせてください。

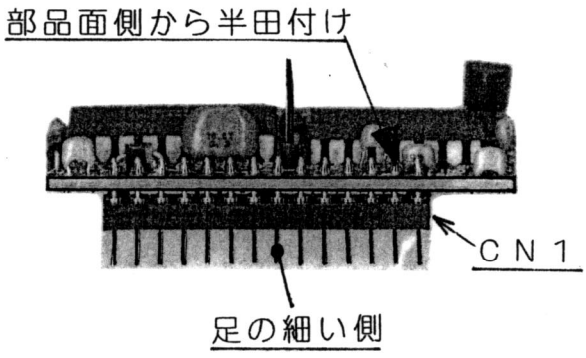
②セラロック、ＩＣをとりつけます。
　セラロックは３本足のまんなかがコモン端子ですので、向きはどちらの向きでも同じです。
　ＩＣの取り付け方向は部品配置図を参考に取り付けてください。

③Ｊ１、Ｊ２にピンヘッダをつけます。
　ピンヘッダは、あらかじめ２Ｐづつに切り離してください。
　ピンヘッダは取り付け穴の直径がぎりぎりの太さのため、入れにくい場合があります。その場合トントンとたたいて、いれてください。

④ここまで半田付けを終了したところで、ハンダ面側のリードをニッパーで切って短くしてください。
　連結ソケットはハンダ面側から取り付けます。連結ソケットは、良く見ると足の太い側と細い側があります。太い側を基板にさして部品面側から半田付けしてください。
　細い側はＩＣソケットに入る太さです。

■部品配置図■





■ＣＮ１接続表■

ＣＮ１	機能	ＣＮ１	機能
１	ＮＣ（無接続）	２８	ＧＮＤ
２	ＮＣ（無接続）	２７	ＴＸ１（２３２）
３	Ｖ－ＩＮ（電源＋）	２６	ＲＸ１（２３２）
４	ＲＸＤ２（２３２）	２５	ＲＸ１（ＴＴＬ）
５	ＴＸＤ２（２３２）	２４	ＴＸ１（ＴＴＬ）
６	ＲＡ２	２３	ＲＡ１
７	ＲＡ３（ＴＸＤ２）	２２	ＲＡＯ
８	ＲＴＣＣ（ＲＸＤ２）	２１	ＯＳＣ１（書込時使用）
９	ＭＣＬＲ	２０	ＯＳＣ２（書込時使用）
１０	ＧＮＤ	１９	ＶＤＤ（５ＶＯＵＴ）
１１	ＲＢＯ	１８	ＲＢ７
１２	ＲＢ１	１７	ＲＢ６
１３	ＲＢ２	１６	ＲＢ５
１４	ＲＢ３	１５	ＲＢ４

■回路の説明■　回路図を参考にお読みください。

①ＣＮ１

外部に接続するコネクタです。２８ピンＩＣの形をしています。それぞれの機能はＣＮ１接続表をごらんください。

②ＳＸ１８ＡＣ（Ｕ１）

ＳＸ１８ＡＣ全ピンがＣＮ１に接続されています。ＯＳＣ１，ＯＳＣ２，ＭＣＬＲは基板内で各部品に接続されています。ＲＡ３，ＲＴＣＣはＪ１、２で２３２に接続出来るようになっています。接続せずにそのまま通常のＩ／Ｏとして、使用する事もできます。ＲＡＯ－２，ＲＢＯ－７はそのままＣＮ１に接続されています。

③電源

電源はＣＮ１－３がＶ－ＩＮ，ＣＮ１－２８がＧＮＤです。電源電圧入力は５～１２Ｖです。レギュレータＳ８１３５０は自己のドロップが０．０３Ｖと小さい為システム全体が５Ｖの場合でも、そのままＶ－ＩＮに５Ｖを入力して動作します。

④発振子

５０ＭＨｚセラロックがついています。コンデンサ内蔵ですので、このままで５０ＭＨｚ動作します。

⑤ＲＳ２３２

ＲＳ２３２は送信２ＣＨ、受信２ＣＨ分あります。１ＣＨ分はＪ１、Ｊ２でＳＸ１８のＲＡ３，ＲＴＣＣに接続できるようになっています。他の１ＣＨ分は入出力とも、ＣＮ１に出ていますので必要にあわせて、接続してください。

⑥リセット

ＭＣＬＲピンが抵抗でプルアップされ、電源リセットが働くようになっています。外部にリセット回路を接続することもできます。

■ＳＸ１８のプログラム開発と書き込み■

ＳＸ１８シリーズのプログラム開発には、専用のアセンブラと、専用のライターが必要です。PARALLAX社　SX-KEY DEVELOPMENT PACKAGE等が使用できます。
（秋月電子通商では、平成１０年８月現在　開発中です。）
ＳＸ１８はＰＩＣ５４シリーズと同様なパラレル書き込みとＯＳＣ１，２を使用したシリアル書き込みの２種類があります。
このキットでは、ＯＳＣ１，２を使用するシリアル書き込みに対応しています。

●ライターとの接続●

書込ライターが、キット基板側からの５Ｖ（ＶＤＤ）をＶＰＰ用電源として使用することは、キット基板の電流が不足するため、できません。
その場合には、別の５Ｖを外部から、ＶＤＤ（ＣＮ１－１９）と書込ライターに供給してください。（SX-KEY等）
ライターが自己の電源で５Ｖ、ＶＰＰを供給する場合は、別の５Ｖを用意する必要はありません。

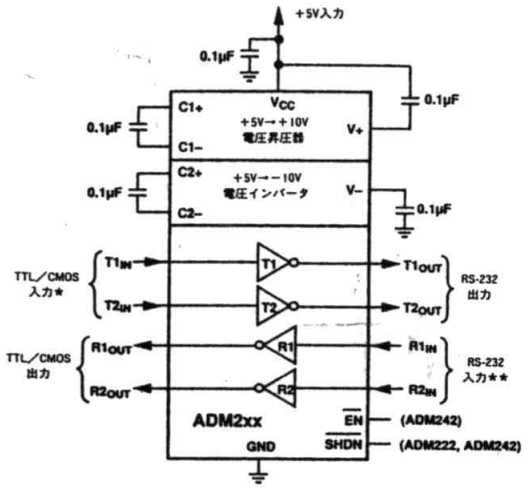
ＳＸ１８キット		ライター
ＣＮ１－２１（ＯＳＣ１）	――――	ＯＳＣ１（ＶＰＰ）
ＣＮ１－２０（ＯＳＣ２）	――――	ＯＳＣ２（データ）
ＣＮ１－１９（ＶＤＤ）	――――	５Ｖ
	└――（５Ｖ別電源）	
ＣＮ１－１０（ＧＮＤ）	――――	ＧＮＤ

（その他のピン（Ｖ－ＩＮ）等はすべて無接続）

ANALOG DEVICES ADM232AAN

高速、+5V、0.1μF CMOS RS-232ドライバ／レシーバ

機能ブロック図



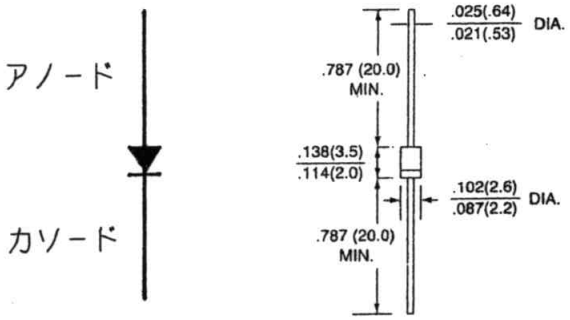
★各TTL／CMOS入力上に400kΩのプルアップ抵抗を内蔵
★★各RS-232入力上に5kΩのプルダウン抵抗を内蔵

特長

200kB/秒の転送レート
小容量（0.1μF）値のチャージ・ポンプ用コンデンサ
＋5V単一電源動作
EIA-232-EおよびV.28規格に適合
2個のドライバと2個のレシーバ
DC-DCコンバータを内蔵
＋5V電源で±9Vの出力振幅
±30Vのレシーバ入力レベル
MAX222／MAX232A／MAX242とピン・コンパチブル

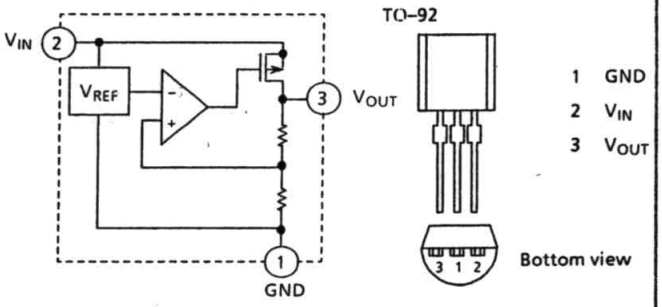
1S10 ショットキーダイオード

VOLTAGE 100 Volts
CURRENT - 1.0 Ampere



高精度ボルテージレギュレータ S81350HG

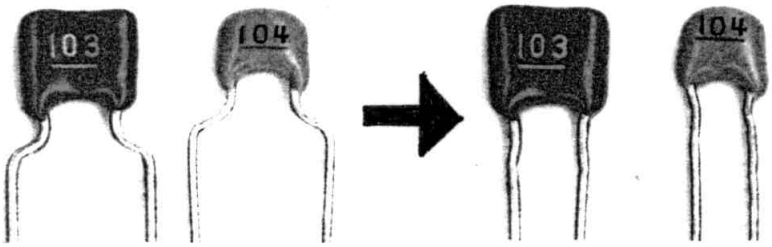
入出力電圧差が小さい
S-81350HG :0.12 V typ. I_{OUT}=40 mA)



積層セラミックコンデンサ

０．１μＦ　（表示１０４）
０．０１μＦ（表示１０３）

あらかじめリードを真っすぐにのばし、２．５ｍｍピッチになるようにする。



1．1 序論

SX シリーズは、FLASH メモリ内蔵の、ハイ・パフォーマンスの 8 ビットシングルチップマイクロコントローラです。最新のデザイン技術とプロセステクノロジーを用い、フルスタティック CMOS で、50MHz オペレーションを実現しました。
RISC ライクなアーキテクチャーにより、わずか 43 の命令セットでプログラム可能で、ターボモード時には、プログラム分岐（3 サイクル）を除き、全てシングルサイクルにて命令を実行します。

1．2 主な特長

- 50MIPS パフォーマンス（@50MHz）
- シングルサイクル命令実行
- 1 万回書込み可能な 2048 x 12 ビット Flash プログラムメモリ
- In-system プログラミング
- シングルステップとブレークポイントデバッグ
- 4MHz RC 発振回路内蔵
- 選択可能なクロックオプション
- 外部発振回路
- 水晶振動子オプション
- アナログコンパレータ（RB0 出力、RB1 入力−、RB2 入力+ 各ピンを利用）
- ブラウンアウト検出機能内蔵（on/off, 4.2V）
- マルチ入力ウェイクアップ機能(MIWU)内蔵（8 ピン）
- シンク／ソース電流 30mA（全出力）
- 1998 UL のサポート 信頼性の高い内部バス
- 統合開発環境（SX-Key）
  エミュレータケーブル
  シングルステップ、リアルタイム（ブレイクポイント）デバッグ機能、エディタ

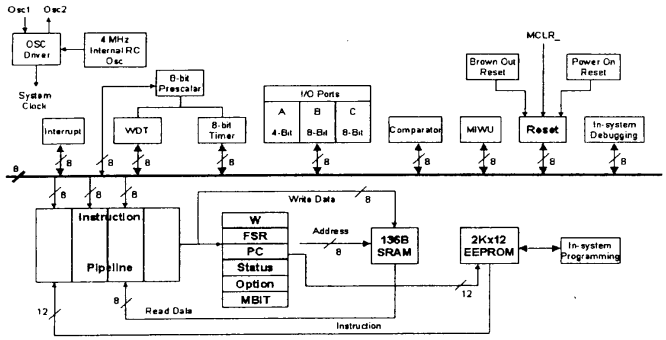


Figure 1.1: SX アーキテクチャ・ブロックダイアグラム

1．2．1 CPU 部の特長

- 20ns 命令サイクル（50MHz 動作時）
- スタティックデザイン（DC～50MHz 動作）
- 43 命令
- 8 レベル・ハードウェアスタック選択可能
- 一定割り込み応答時間（60ns 内部、100ns 外部@50MHz）
- 割り込み時の PC, W (Work), STATUS, FSR の各レジスタのセーブ／リストア
- W レジスタからの RTCC の扱いが容易

1．2．2 周辺及び I/O 部の特長

- プログラムにより入出力選択可能（各ピン）
- TTL/CMOS レベル選択可能（入力ピン）
- 内部プルアップ選択可能（全ピン：～20kΩ to VDD）
- ポート B, C 入力をシュミットトリガーとして選択可能
- シンク／ソース電流 30mA（全出力）
- シンメトリカル・ドライブ（RA 出力：Vdrop +/−）

1．3 アーキテクチャ

他のマイクロコントローラに類を見ない SX シリーズの高いパフォーマンスは、卓越した特長と機能により実現しています。エンハンスト・フェッチ−デコード−実行−ライトバック・パイプラインは、SX に 1 命令 1 クロックで命令を実行させ、50MHz 動作時には２０ｎｓの命令サイクルを実現しました。
SX シリーズはハーバードアーキテクチャのモデルに基づき開発されており、このアーキテクチャでは、プログラムコードとデータ値を個別のメモリ領域に格納して、異なったバス経由で転送し最大限のバンド幅を確保しています。
SX28/20/18AC シリーズは、2K x 12bit の内部 Flash プログラムメモリと 136 バイトの RAM を持っています。RAM は直接または間接にアドレッシング可能です。全てのファンクション・レジスタは、データメモリ（W レジスタを含む）にマップされます。
ＡＬＵは、８ビット幅で、算術演算と論理演算の能力を持ちます。
Wレジスタは、ＡＬＵ用のワークレジスタです。Wレジスタは通常２オペランド命令の内の１オペランドを保持し、命令の実行に従って、ALU は、STATUS レジスタの Carry (C), Zero (Z), and Digit Carry (DC) フラグの値に影響します。
SX シリーズは、外部回路コストを削減するために各種機能を内蔵しています。
パワー・オン・リセット（POR）とデバイス・リセット・タイマーは、外部リセット回路を必要としません。
また、内蔵の４ＭＨｚクロックを持っており、外部発振回路を含めると５種類の発振回路から選択可能です。
消費電流を抑えるスリープモード、ウォッチドッグタイマやコードプロテクト機能を持ち、トータルシステムコストを減らすことができます。

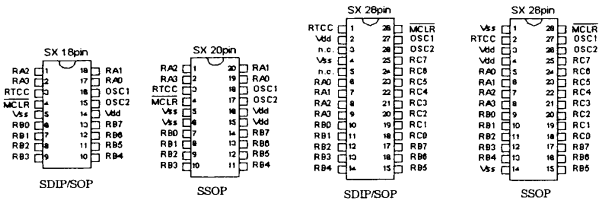
1．4 プログラミングとデバッグ

SX シリーズは、Flash メモリを内蔵することにより紫外線イレーサー等を不要にし、コード開発に理想的な環境を提供します。
SX シリーズではボンドアウトチップを利用した一般的なエミュレータで発生する問題を防止するために、エミュレーションチップに必要な埋め込みフックを持たせました。これによりエンベッデドシステムに要求される信頼性と低コストエミュレーション・ソリューションを実現しました。
SX シリーズでは、開発を効率よく行うために SX-Key と呼ぶ統合開発環境（エディタ、マクロアセンブラ、プログラマとデバッガ、エミュレータ・ケーブルを含む）を準備しております。

2-1 命令セット

ニーモニック，オペランド	解説	サイクル（標準）	サイクル（ターボ）	オペコード	フラグへの影響
論理演算命令群					
AND fr, W	frに対してWでAND演算を行なう。(fr = fr & W)	1	1	0001 011f ffff	Z
AND W, fr	Wに対してfrでAND演算を行なう。(W = W & fr)	1	1	0001 010f ffff	Z
AND W, #literal	Wに対してliteral(文字)でAND演算を行なう。(W = W & lit)	1	1	1110 kkkk kkkk	Z
NOT fr	frの1の補数を取る。(fr = fr ^ FF)	1	1	0010 011f ffff	Z
OR fr, W	frに対してWでOR演算を行なう。(fr = fr | W)	1	1	0001 001f ffff	Z
OR W, fr	Wに対してfrでOR演算を行なう。(W = W | fr)	1	1	0001 000f ffff	Z
OR W, #literal	Wに対してliteral(文字)でOR演算を行なう。(W = W | lit) 結果が0であれば、Zは1にセットされ、さもなければ0にクリアされる。	1	1	1110 kkkk kkkk	Z
XOR fr, W	frに対してWでXOR演算を行なう。(fr = fr ^ W)	1	1	0001 101f ffff	Z
XOR W, fr	Wに対してfrでXOR演算を行なう。(W = W ^ fr)	1	1	0001 100f ffff	Z
XOR W, #literal	Wに対してliteral(文字)でXOR演算を行なう。(W = W ^ lit)	1	1	1111 kkkk kkkk	Z
算術及びシフト演算演算命令群					
ADD fr, W	Wがfrに加算される。(fr = fr + W) FUSEXレジスタのCFビットが0であれば、Cが結果に加算され	1	1	0001 111f ffff	C, DC, Z
ADD W, fr	frがWに加算される。(W = W + fr)	1	1	0001 110f ffff	C, DC, Z
CLR fr	frが0にクリアされる。(fr = 0)	1	1	0000 011f ffff	Z
CLR W	Wが0にクリアされる。(W = 0)	1	1	0000 0100 0000	Z
CLR !WDT	ウォッチドッグタイマがクリアされる。割り当てられていれば、プリスケーラも一緒にクリアする。TOとPDは1にセットされる。	1	1	0000 0000 0100	TO, PD
DEC fr	frがデクリメントされる。(fr = fr − 1)	1	1	0000 111f ffff	Z
DECSZ fr	frがデクリメントされる。(fr = fr − 1) frが0であれば、次の命令はスキップされる。	1 or 2 (skip)	1 or 2 (skip)	0010 111f ffff	−
INC fr	frがインクリメントされる。(fr = fr + 1)	1	1	0010 101f ffff	Z
INCSZ fr	frがインクリメントされる。(fr = fr + 1) frが0であれば、次の命令はスキップされる。	1 or 2 (skip)	1 or 2 (skip)	0011 111f ffff	−
NOP	何もしない。	1	1	0000 0000 0000	−
RL fr	frに対して左ローテートを行う。入口側は、CはLSBの中にシフトされて値を保持する。出口側は、Cは前回のfrのMSBの値を保持する。	1	1	0011 011f ffff	C
RR fr	frに対して右ローテートを行う。入口側は、CはMSBの中にシフトされて値を保持する。出口側は、Cは前回のfrのLSBの値を保持する。	1	1	0011 001f ffff	C
SUB fr, W	frからWが減算される。(d = 1) FUSEXレジスタのCF_ビットが0であれば、Cが結果に加算される。アンダーフローが起これば、C = 0。アンダーフローが起こらなければ、C = 1。最小ニブルアンダーフローが起これば、DC = 0。最小ニブルアンダーフローが起こらなければ、DC = 1。	1	1	0000 101f ffff	C, DC, Z
SWAP fr	frの上位と下位数のニブルを入れ換える。	1	1	0011 101f ffff	−
ビット操作命令群					
CLRB bit	ビットが0にクリアされる。(fr.bit = 0)	1	1	0100 bbbf ffff	−
SB bit	bit = 1の場合、次の命令をスキップする。	1 or 2 (skip)	1 or 2 (skip)	0111 bbbf ffff	−
SETB bit	ビットが1にセットされる。	1	1	0101 bbbf ffff	−
SNB bit	bit = 0の場合、次の命令をスキップする。	1 or 2 (skip)	1 or 2 (skip)	0110 bbbf ffff	−
データ転送命令群					
MOV fr, W	frに対してWを転送する。(fr = W)	1	1	0000 001f ffff	−
MOV W, fr	Wに対してfrを転送する。(W = fr)	1	1	0010 000f ffff	Z
MOV W, fr−W	frからWを減算して、結果をWに転送する。(d = 0) FUSEXレジスタのCF_ビットが0であれば、Cが結果に加算される。アンダーフローが起これば、C = 0。アンダーフローが起こらなければ、C = 1。最小ニブルアンダーフローが起これば、DC = 0。最小ニブルアンダーフローが起こらなければ	1	1	0000 100f ffff	C, DC, Z
MOV W, #lit	Wに対してliteral(文字)を転送する。(W = lit)	1	1	1100 kkkk kkkk	−
MOV W, /fr	frの１の補数をWに転送する。(W = fr ^ FF)	1	1	0010 010f ffff	Z
MOV W, −−fr	fr − 1がWに対してロードされる。(W = fr − 1)	1	1	0000 110f ffff	Z
MOV W, ++fr	fr + 1がWに対してロードされる。(W = fr + 1)	1	1	0010 100f ffff	Z
MOV W, <<fr	frの左ローテートした値をWに転送する。入口側は、CはLSBの中にシフトされて値を保持する。出口側は、Cは前回のfrのMSBの値を保持する。	1	1	0011 010f ffff	C
MOV W, >>fr	frの右ローテートした値をWに転送する。入口側は、CはMSBの中にシフトされて値を保持する。出口側は、Cは前回のfrのLSBの値を保持する。	1	1	0011 000f ffff	C
MOV W, <>fr	frの入れ換えたニブル値をWに転送する。	1	1	0011 100f ffff	−
MOV W, !MODE	Wに対してモードビットを読み出す。	1	1	0000 0100 0010	−
MOVSZ W, −−fr	fr − 1がWに対してロードされる。(W = fr − 1) frが0であれば、次の命令はスキップされる。	1 or 2 (skip)	1 or 2 (skip)	0010 110f ffff	−
MOVSZ W, ++fr	fr + 1がWに対してロードされる。(W = fr + 1) frが0であれば、次の命令はスキップされる。	1 or 2 (skip)	1 or 2 (skip)	0011 110f ffff	−
MOV !MODE, W	モードに対してWを書き出す。	1	1	0000 0100 0011	−
MOV !OPTION, W	OPTIONにWを書き出す。(OPTION = W)	1	1	0000 0000 0010	−
TEST fr	frに対してfrを転送する。(fr = fr) frが0の時は、テストに使う。	1	1	0010 001f ffff	Z
制御移動命令群					
CALL addr8	PC<10:0> + 1 -> Stack addr8 ————> PC<7:0> PA1:PA0 ————> PC<10:9>	2	3	1001 kkkk kkkk	−
JMP addr9	PC<8:0> <— addr9 PC<10:9> <— PA1:PA0	2	3	101k kkkk kkkk	−
RET	Wの影響無しで制御を戻す。	2	3	0000 0000 1100	−
RETP	RETと同等。しかし、PA1:PA0の中の戻りアドレスビット10:9を書く。	2	3	0000 0000 1101	PA1:PA0
RETI	割り込みから戻る。(pop W, STATUS, FSR, PC)	2	3	0000 0000 1110	C, DC, Z
RETW	RETIと同等。プリスケーラとディレイの影響無しでRTCCからWを減算する。	2	3	0000 0000 1111	C, DC, Z
RETW #literal	Wの中にliteral(文字)を置いて戻る。	1	3	1000 kkkk kkkk	−
システム制御命令群					
BANK n	FSR7:FSR5ビットにnを書き出す。(n = 0-7)	1	1	0000 0001 1nnn	−
IREAD	MODE:Wの中のアドレス[MODE:W]に命令を読み出	1	4	0000 0100 0001	−
MODE n	MODEにnを書き出す。	1	1	0000 0101 nnnn	−
PAGE n	PA2:PA0ビットにnを書き出す。(n = 0-3)	1	1	0000 0001 0nnn	−
SLEEP	ウォッチドッグタイマがクリアされ、発振器は停止する。TOが1にセットされる。PDが0にクリアされる。	1	1	0000 0000 0011	TO, PD
TRIS fr	frのTRISレジスタ(fr = 5, 6, or 7)に対してWがコピーされる。Wのあるビットの"1"は、対応するポートのピンの出力バッファを禁止したり、入力を許可する。"0"ビットである間は、出力バッファ(ハイあるいはロウ出力)を許可する。	1	1	0000 0000 0fff	−

３．１ ピン配置図



ピン解説

ピン名	ピンタイプ	入力レベル	解説
RA0	I/O	TTL/CMOS	双方向I/Oピン
RA1	I/O	TTL/CMOS	双方向I/Oピン
RA2	I/O	TTL/CMOS	双方向I/Oピン
RA3	I/O	TTL/CMOS	双方向I/Oピン
RB0	I/O	TTL/CMOS/ST	双方向I/Oピン；コンパレータ出力；MIWUモード
RB1	I/O	TTL/CMOS/ST	双方向I/Oピン；コンパレータ負入力；MIWUモード
RB2	I/O	TTL/CMOS/ST	双方向I/Oピン；コンパレータ正入力；MIWUモード
RB3	I/O	TTL/CMOS/ST	双方向I/Oピン；MIWUモード
RB4	I/O	TTL/CMOS/ST	双方向I/Oピン；MIWUモード
RB5	I/O	TTL/CMOS/ST	双方向I/Oピン；MIWUモード
RB6	I/O	TTL/CMOS/ST	双方向I/Oピン；MIWUモード
RB7	I/O	TTL/CMOS/ST	双方向I/Oピン；MIWUモード
RC0	I/O	TTL/CMOS/ST	双方向I/Oピン
RC1	I/O	TTL/CMOS/ST	双方向I/Oピン
RC2	I/O	TTL/CMOS/ST	双方向I/Oピン
RC3	I/O	TTL/CMOS/ST	双方向I/Oピン
RC4	I/O	TTL/CMOS/ST	双方向I/Oピン
RC5	I/O	TTL/CMOS/ST	双方向I/Oピン
RC6	I/O	TTL/CMOS/ST	双方向I/Oピン
RC7	I/O	TTL/CMOS/ST	双方向I/Oピン
RTCC	I	ST	リアルタイムクロック／カウンタ入力
MCLR	I	ST	マスタクリア(リセット)入力 アクティブロウ
OSC1/In/Vpp	I	ST	水晶発振入力／外部クロックソース入力
OSC2/Out	O	CMOS	内部RCモードでは、Vddにプルアップする。
Vdd	P	-	正電源
Vss	P	-	グランド

メモ：I = 入力，O = 出力，I/O = 入出力，P = 電源，- = 未使用，TTL = TTL 入力，CMOS = CMOS 入力，ST = シュミットトリガー入力，MIWU = マルチ入力ウェークアップ入力

３．２ パイプライン解説

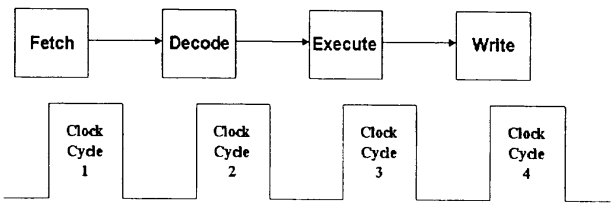


Figure 3.1: パイプラインとクロックの仕組み

SX シリーズのパイプラインにおいては Figure 3.1 にあるように、命令を実行するまでに 4 つの段階があります。
最初のクロックサイクルにおいて最初の命令がメモリからフェッチされます。2 番目のクロックサイクルで、最初の命令はデコードされ、2 番目の命令がフェッチされます。3 番目のクロックサイクルでは、最初の命令が実行され、2 番目の命令はデコードされ、また 3 番目の命令がフェッチされます。4 番目のクロックサイクルでは、最初の命令の結果が処理先に書き出されます。そして、2 番目の命令は実行され、3 番目の命令はデコードされ、また 4 番目の命令がフェッチされます。5 番目のクロックサイクルでは、もと図の通りです。パイプラインが一杯になった場合には、命令はクロック毎に実行されます。
プログラムカウンタ内部の値を直接変更する命令（例えば、jumps, calls, 等々）においては、パイプラインはクリアされるので後で再度入れ直す必要があります。パイプラインがクリアされた時、フェッチとデコードのステップへは、nop 命令が書き込まれ、その結果、効率よく無効命令は破棄されます。

３．２．１ Read-Modify-Write について

I/O ポートに対して連続的に SETB または CLRB を実行している場合に、命令に使われる入力データは、命令の実行中は有効でなければなりませんし、出力結果は、命令実行完了後、有効となります。
このため、一般的に高速で動作している時、I/O ポートピンにおける連続的な Read-Modify-Write の動作においてエラーが起きる可能性があります。
SX では、このようなデータエラーを防ぐため内部に Write-Back セクションを持っていますが、高速なクロックを用いて同一の I/O ポートピンに対して連続的な Read-Modify-Write の動作を行う場合には、"nop"命令を挿入するようにして下さい。
I/O ポートの読み出しとは、実際にピンにアクセスしており、出力データをラッチしているわけではありません。つまり、もしピンの出力ドライバが許可されて、HIGH にドライブされていたとしても、外部システム側が LOW に保持していていれば、ポートの読み出し値は LOW となります。

３．３ デバイスコンフィグレーションレジスタ

SX には 3 個のレジスタがあり、ターボモードの設定、拡張スタック（8 レベルの深さ）の設定、内部ＲＣ発振回路等の設定を行います。
FUSE 及び FUSEX レジスタは、通常の動作中にプログラムから設定することはできません。
これらのレジスタはプログラム書き込み時にのみ設定可能です。

DEVICE レジスタは、リードオンリーのハートワイヤードで変更できません。

３．３．１ FUSE Word (Read/Program at 1FFFh in main memory map)

11	10	9	8	7	6	5	4	3	2	1	0
TURBO_	SYNC_	OPTIONX_	STACKX_	IRC_	DIV2	DIV1	DIV0	CP_	WDTE	FOSC1	FOSC0

TURBO_ - ターボモード許可
0 = turbo（命令クロック = OSC/1）
1 = normal（命令クロック = OSC/4）

SYNC_ - 同期入力許可（ターボモード用）:
0 = 許可
1 = 禁止

OPTIONX_ - OPTION レジスタ拡張許可:
0 = OPTION レジスタを 6 ビットから 8 ビットに拡張（RTCC_OF_と RTCC_IE_用）
1 = OPTION レジスタを標準 6 ビットにします（MSB2 ビットは 1 にセット）

STACKX_ - スタック拡張許可:
0 = 8 レベル
1 = 2 レベル

IRC_ - 内部 RC 発振回路許可:
0 = 許可 - OSC1 はプルダウン、OSC2 はプルアップします。
1 = 禁止 - OSC1 と OSC2 は、FOSC1: FOSC0 に従って動作します。

DIV2: DIV0 - 内部 RC 発振用分周:
000b = 4 MHz
001b = 2 MHz
010b = 1 MHz
011b = 500 KHz
100b = 250 KHz
101b = 125 KHz
110b = 62.5 KHz
111b = 31.25 KHz

CP_ - コードプロテクト許可:
0 = 許可（FUSE，コードと ID メモリは 3-nibbles-XOR'd を読み出す）
1 = 禁止

WDTE - ウォッチドッグタイマ許可:
0 = 禁止
1 = 許可

FOSC1: FOSC0 - 外部発振器設定（IRC = 1 の時、有効）:
00b = LP - 低電圧水晶振動子
01b = HS - 高速水晶振動子
10b = XT - 一般の水晶振動子
11b = RC - RC 発振器。OSC2 は、プルアップします。(CLKOUT 出力なし)

３．３．２ FUSEX Word (Read/Program via programming command)

11	10	9	8	7	6	5	4	3	2	1	0
Preset	Preset	Preset	Preset	B_OSC	CF_	BOR1	BOR0	RAM1	RAM0	MEM1	MEM0

Presets - 工場設定値。これらのビットは、変更しないで下さい。

B_OSC - 8 クロック分、OSC2 を Low にすることによる発振器の停止を許可します。

CF_ - アクティブロウ - ADD と SUB 命令に対してキャリーフラグ入力を入れる。

BOR1: BOR0 - 工場設定値

RAM1: RAM0 - チップ上に構成されるの RAM バンク数:
00b = 1
01b = 2
10b = 4
11b = 8 banks

MEM1: MEM0 - チップ上に構成されるメモリサイズ:
00b = 512
01b = 1024
10b = 2048
11b = 4096 words

３．３．３ DEVICE Word (Hardwired read-only)

11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	res.	res.	res.	res.	RAM1	RAM0	MEM1	MEM0

res. - 工場設定値。

RAM1: RAM0 - チップ上の絶対 RAM バンク数:
00b = 1
01b = 2
10b = 4
11b = 8 banks

MEM1: MEM0 - チップ上の絶対メモリサイズ:
00b = 512
01b = 1024
10b = 2048
11b = 4096 words

3．4 スペシャルファンクションレジスタ

スペシャルファンクションレジスタは、CPU制御とデバイス操作で使われるレジスタです。

Table 3.2 スペシャルファンクションレジスタ

Addr	Name	Function
00h	INDIRECT	間接アドレッシングに使われる
01h	RTCC	リアルタイムクロック／カウンタ
02h	PC	プログラムカウンタ（low byte）
03h	STATUS	ALUのステータスビットを保持
04h	FSR	ファイルセレクトレジスタ
05h	RA	RAポート制御レジスタ
06h	RB	RBポート制御レジスタ
07h	RC(*)	RCポート制御レジスタ

NOTE(*):SX18パッケージでは通常のRAMとして使用可能

3．4．1 INDIRECT - 間接レジスタ (00h)

このレジスタは、物理的にインプリメントされていませんが、間接アドレッシングを使ってアクセスすることができます。オペランドとして INDIRECT を使う命令は、FSR の値によって示されたレジスタを対象として処理されます。

3．4．1．1 直接及び間接アドレッシングの例

```
Direct addressing:
mov RA,#01        ;move "1" to RA
Indirect addressing:
mov FSR,#RA       ;FSR = address of RA
mov INDIRECT,#01  ;move "1" to RA
```

NOTE:間接アドレッシングでの INDIRECT の演算の結果は、FSR で指し示したレジスタを対象に実行された結果で得られます。

3．4．2 RTCC (01h) リアルタイムクロック／カウンタとWレジスタ

RTCCは8ビットのリアルタイムタイマ／カウンタです。RTCCレジスタは、あらゆる命令サイクル（プリスケーラなし）毎にインクリメントされます。
カウンタモードでは、RTCCは、RTCCピンの入力サイクル（プリスケーラあり）毎にインクリメントされます。
プリスケーラは、RTCCまたはウォッチドッグタイマを16ビットまでを有効長にするために使われます。OPTIONレジスタ bit 7のRTWビットの設定によって、レジスタ01hの読み出しは、RTCCを読むこと（RTWは1）またはWを読むこと（RTWは0）となります。
なお、RTCCの値はRTWビット設定にかかわらず残ります。RTCCとプリスケーラの詳細につきましては5.2を参照して下さい。

3．4．3 PC - プログラムカウンタレジスタ(02h)

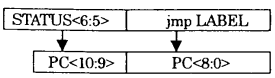
プログラムカウンタレジスタは、プログラムカウンターの下位8ビットを保持するレジスタです。
ジャンプ命令等の実行を行うためにランタイムでアクセス可能です。命令が実行された時は常にPCには行き先がロードされ、STATUSレジスタの上位ビットは、プログラムカウンターの上位バイトにロードされます。これはコードメモリのページ境界を越えて計算されたジャンプやサブルーチンコールへの命令に必要です。

3．4．3．1 プログラムカウンタ

プログラムカウンターは、実行される命令の11ビットアドレス（コードメモリ）です。下記で、プログラムカウンタとSTATUSレジスタの動作について説明します。

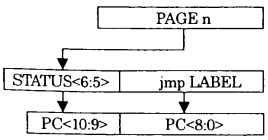
3．4．3．2 ジャンプの解説

JMP命令が実行される時、プログラムカウンタの下位9ビットには、指定されたラベルのアドレスがロードされます。また、プログラムカウンタの上位2ビットには、STATUSレジスタからページセレクトビット（PA1:PA0）がロードされます。
従って、ジャンプ命令の実行前にページセレクトビットが正しいページを指し示しているかどうか注意して下さい。



3．4．3．3 ページジャンプの解説

JMP命令の実行において、その行き先が異なるページにある時、ジャンプ命令の前に指定したページを指し示すためにページセレクトビットをセットしなければなりません。
この操作は、SETBとCLRB命令あるいはSTATUSレジスタへ値を書くことで確実に実行されます。
SXシリーズでは、PAGEと呼ばれるシングルサイクルで実行する新しい命令を持っています。
PAGEジャンプの使い方の例は以下の通りです。



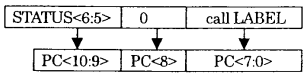
Note: nは、0から3の間の数字でなければなりません。

3．4．3．4 コールについて

CALL命令により次のような命令が実行されます。

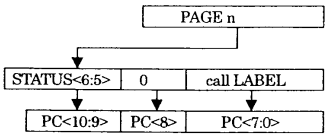
- プログラムカウンタの現在値は、インクリメントされてスタックの一番上にプッシュされます。
- ラベルのアドレスの下位8ビットは、プログラムカウンタの下位8ビットにコピーされます。
- PCの9番目のビットは、クリアされます。
- STATUSレジスタのページセレクトビットは、PCの上位2ビットにコピーされます。

上記で説明されていることは、呼び出し先の先頭アドレスはプログラムコード空間の、ページの前半分に置かなければならないことを意味します。例えば、00h-0ffh, 200h-2ffh, 400h-4FFh等々。



3．4．3．5 ページコールについて

異なるページに存在するサブルーチンを呼ぶ必要がある時、呼び出しを実行する前に、指定されたページを指し示すためにページセレクトビットをセットしなければなりません。
この操作は、SETBやCLRB命令の実行や、STATUSレジスタへの値を書くことで確実に実行されます。
SXでは、ページの呼び出しをシングルクロックサイクルで実行する新しい命令を持っています。ページの呼び出しの使い方の例は以下の通りです。



Note: nは、0から3の間の数字でなければなりません。

3．4．3．6 サブルーチンからのリターン

サブルーチンは、通常リターンタイプの命令で終了します。いくつかのリターン命令の説明の前に、スタックの動作について説明します。

3．4．4 スタック

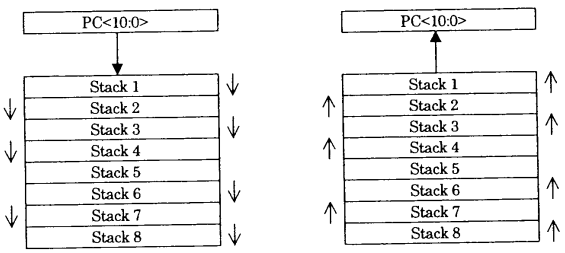
スタックは、サブルーチンが終了してから戻る場所を覚えておくために使われるメモリ空間です。スタックには8レベルの深さがあり、8レベルまでネストされたサブルーチンの戻りアドレスを覚えておくことができます。ネストされたサブルーチンとは、他のサブルーチンの内部から呼び出されたサブルーチンのことです。スタックには、pushとpopの2つの機能があります。例えると、スタックは、サラダビュフェのプレートホルダーと同じように動作します。pushは、スタック上にプレートを置くのと同じであり、popはスタックからプレートを取り出すのと同じような動作です。

3．4．4．1 プッシュ

サブルーチンが呼ばれる時、戻りアドレスはスタックにプッシュされます。
この時、スタックの各々のアドレスは、ストアされる新しいアドレス用の空間を確保するために次の下位レベルに移動します。スタック1には、プログラムカウンタにあったアドレスが入ります。スタック8は、スタック7にあったアドレスで上書きされ、スタック8にあった内容は消去されます。

3．4．4．2 ポップ

リターン命令が実行される時、スタックにポップされます。
スタック1のアドレスは、プログラムカウンタにコピーされ、スタックの各々のアドレスは、次の上位レベルに移動します。スタック1には、スタック2にあった値が入り、スタック7にはスタック8の内容が上書きされ、その結果として、スタック7とスッタク8の内容は同じになります。



3．4．4．3 リターン

SXには5つの異なるリターン命令があります。
RET(Return)命令は、呼び出し元のCALL命令に続く命令にプログラムカウンタをセットするために単純にスタックをポップしています。
RETW命令（Wレジスタへの値を持って戻る）は、RETと同じよう動作しますが、指定された値をWレジスタにロードします。
RETP命令は、スタックをポップして、元のページに戻るために指し示したページセレクタビットを更新します。
RETI命令は、スタックをポップして、加えて割り込みハンドリング中にシャドウレジスタへ保持されていたW, STATUSとFSRを元に戻すために使われます。
RETIWは、RETIと同じように動作しますが、加えてRTCCのWレジスタの2の補数値を加えることによって、RTCCを補正する処理を行います。

3．4．5 STATUS レジスタ (03h)

このレジスタは、ALU での演算結果やページセレクトビットとリセットステータスを保持します。STATUS レジスタは、いつでもアクセス可能ですが、リセットビットの PD と TO はリードオンリーです。
STATUS レジスタに書き出す場合、ALU ステータスビットは書き出し処理の終了時点で書き換わります。その結果、意図したのと異なる結果で STATUS レジスタに残ります。そのことに注意して下さい。
従って、このレジスタに対しては SETB と CLRB 命令のみを実行するようにして下さい。

7	6	5	4	3	2	1	0
PA2	PA1	PA0	TO	PD	Z	DC	C

Bit 7-5: (*) ページセレクトビット PA2:PA0
000 = Page 0 (000h - 01FFh)
001 = Page 1 (200h - 03FFh)
010 = Page 2 (400h - 05FFh)
011 = Page 3 (600h - 07FFh)
100 = Page 4 (800h - 09FFh)
101 = Page 5 (A00h - 0BFFh)
110 = Page 6 (C00h - 0DFFh)
111 = Page 7 (E00h - 0FFFh)

NOTE(*): コード空間が 4K 未満のデバイスでは、PA2:PA0 は汎用目的の read/write ビットとして使用されます。

Bit 4: タイムアウトビット, TO.
1 = パワーアップまたは、CLRWDT または SLEEP 命令実行後、"1"にセットされます。
0 = ウォッチドッグタイマのタイムアウト発生
Bit 3: パワーダウンビット, PD.
1 = パワーアップまたは、CLRWDT 命令実行後、"1"にセットされます。
0 = SLEEP 命令によって"0"にセットされます。
Bit 2: ゼロビット, Z.
1 = 算術演算の結果が"0"
0 = 算術演算の結果がゼロでない
Bit 1: デジットキャリービット, DC.
加算後:
1 = 下位 4 ビットからのキャリー発生。
0 = 下位 4 ビットからのキャリーなし。
減算後:
1 = 下位 4 ビットへのボローなし。
0 = 下位 4 ビットへのボローあり。
Bit 0: キャリービット, C.
1 = 結果の MSB からキャリーが発生。rotate (RR and RL) 命令では、それぞれロウあるいはハイオーダービットでロードされます。
0 = MSB からキャリーなし

3．4．6 FSR - ファイルセレクトレジスタ (04h)

SX シリーズでは、12 ビットのオペコードを使用します。オペランドとしてレジスタを指定する命令は、5 ビットのレジスタアドレスまでです。つまり、00h から 1Fh までがレジスタとしてアクセスすることが可能です。
FSR は、1Fh を越えたレジスタにアクセスするために使用します。
Figure 3.2 では、FSR の上位 3 ビットが 8 つの RAM バンクのうちの 1 つを選択する方法を説明しています。
RAM レジスタアドレスの 00h - 0Fh の領域は、FSR の内容に関わらず常にアクセスすることができるグローバルな領域です。RC(07h)は、18 ピンの SX において汎用 RAM として利用できます。

以下は、バンク 4 のレジスタ 10h に書き出す方法の例です。

```
mov W,#80h ;Select Bank 4 by
mov FSR,W  ;setting FSR<7:5>
mov W,#64h ;load register 10h with
mov 10h;W  ;the literal 100d
```

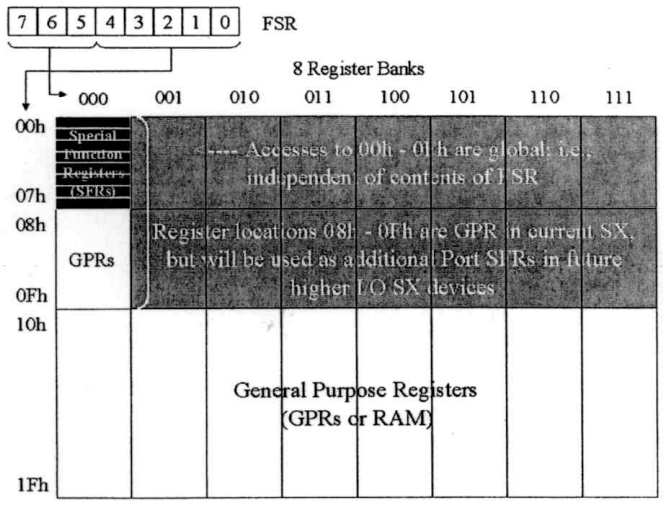


Figure 3.2: FSR と RAM への関係或いはデータメモリマップ

3．4．6．1 直接アドレッシング

要求されたレジスタへ確実に書き出すためには、FSR へ正しい値を最初に書かなければなりません。次のテーブルとプログラムはバンクされたレジスタに直接アクセスする方法を示します。

If you want bank...	Value to move to FSR is...
0	000h
1	030h
2	050h
3	070h
4	090h
5	0B0h
6	0D0h
7	0F0h

```
mov W,#070h ;Select RAM Bank 3
mov FSR,W
clr 010h    ;Clear register 10h on Bank3
mov W,#0D0h ;Select RAM Bank 6
mov FSR,W
clr 010h    ;Clear register 10h on Bank6
```

3．4．6．2 間接アドレッシング

間接アドレッシングを通してあるレジスタにアクセスするためには、FSR にアクセスしたいレジスタの 8 ビットアドレスを設定して、オペランドとして INDF を使用します。
この例では、間接アドレッシングを使い、あらゆるバンク上のあらゆる RAM レジスタをクリアします。

```
Temp = 008h
INIT
mov W,Temp
mov FSR,W ;FSR = TEMP addr
LOOP
clr INDF  ;Clear location
incsz FSR ;Next location
jmp LOOP  ;Repeat until all clear
DONE
jmp DONE  ;Then wait forever
```

3．4．6．3 バンク命令

SX では、バンクの呼び出しをシングルクロックサイクルで実行する命令を持っています。バンク命令は、1 命令サイクルで、FSR のバンクセレクトビットに対して指定のバンクを設定することができます。
下記はバンク命令を使う方法の例です。

```
bank 3   ;Make FSR point to bank 3
clr 10h  ;Clear register 10h (bank 3)
bank 6   ;Make FSR point to bank 6
mov W,#1 ;Set register 10h (bank 6) to 1
mov 10h,W
```

3．5 ポートコントロールレジスタ

3．5．1 ポートAレジスタ

ポートAの I／Oピンを設定するために 3 つのレジスタがあります。RA レジスタは、入力または出力かポートAのデータ方向を設定します。LEVL_A レジスタは、TTL または CMOS 電圧レベルに入力ピンを設定します。PLP_A レジスタは、ポートAピンのプルアップ抵抗を許可／禁止します。これら 3 つのレジスタにアクセスするためには、MODE レジスタに指定された値を最初に書き込まなければなりません。
ポートAにアクセスするために設定する MODE レジスタの値は Table 3.3 を参照して下さい。これらのレジスタのビットは、パワーアップ時に"1"にセットされます。

3．5．1．1 RA - データ方向レジスタ

7	6	5	4	3	2	1	0
				RA3	RA2	RA1	RA0

このレジスタのビットを"1"にセットすることで対応する I／Oポートピンを入力モード（ハイインピーダンス）にセットします。
このレジスタのビットを"0"にセットすることで対応する I／Oポートピンを出力モードにセットします。

3．5．1．2 LVL_A - TTL/CMOS 選択レジスタ

7	6	5	4	3	2	1	0
				RA3	RA2	RA1	RA0

このレジスタのビットを"1"にセットすることで対応する I／Oポートピンの入力レベルをＴＴＬにセットします。
このレジスタのビットを"0"にセットすることで対応する I／Oポートピンの入力レベルをＣＭＯＳにセットします。

3．5．1．3 PLP_A - プルアップ抵抗許可レジスタ

7	6	5	4	3	2	1	0
				RA3	RA2	RA1	RA0

このレジスタのビットを"0"にセットすることで対応する I／Oポートピンのプルアップ抵抗を許可します。
このレジスタのビットを"1"にセットすることで対応する I／Oポートピンのプルアップ抵抗を禁止します。

3．5．2 ポートBレジスタ

ポートBのI／Oピンを設定するために6つのレジスタがあります。
RBレジスタは、入力か出力かポートBのデータ方向を設定します。
LEVL_Bレジスタは、TTLまたはCMOS電圧レベルに入力ピンを設定します。
PLP_Bレジスタは、ポートAピンのプルアップ抵抗を許可／禁止します。ST_Bレジスタは、ポートB入力ピンのシュミットトリガー入力を許可／禁止します。
これらのレジスタにアクセスするために、MODEレジスタに特定の値を最初に書かなければなりません。
ポートBにアクセスするために設定するMODEレジスタの値はTable 3.3を参照して下さい。これらのレジスタのビットは、パワーアップ時に"1"にセットされます。

3．5．2．1 RB- データ方向レジスタ

7	6	5	4	3	2	1	0
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

このレジスタのビットを"1"にセットすることで対応するI／Oポートピンを入力モード（ハイインピーダンス）にセットします。
このレジスタのビットを"0"にセットすることで対応するI／Oポートピンを出力モードにセットします。

3．5．2．2 LVL_B-TTL/CMOS選択レジスタ

7	6	5	4	3	2	1	0
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

このレジスタのビットを"1"にセットすることで対応するI／Oポートピンの入力レベルをＴＴＬにセットします。
このレジスタのビットを"0"にセットすることで対応するI／Oポートピンの入力レベルをＣＭＯＳにセットします。

3．5．2．3 PLP_B- プルアップ抵抗許可レジスタ

7	6	5	4	3	2	1	0
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

このレジスタのビットを"0"にセットすることで対応するI／Oポートピンのプルアップ抵抗を許可します。
このレジスタのビットを"1"にセットすることで対応するI／Oポートピンのプルアップ抵抗を禁止します。

3．5．2．4 ST_B- シュミットトリガー許可レジスタ

7	6	5	4	3	2	1	0
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

このレジスタのビットを"0"にセットすることで対応するI／Oポートピンのシュミットトリガー入力を許可します。
このレジスタのビットを"1"にセットすることで対応するI／Oポートピンのシュミットトリガー入力を禁止します。

3．5．2．5 WKEN_B- ウェイクアップ許可レジスタ

7	6	5	4	3	2	1	0
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

このレジスタのビットを"1"にセットすることで対応するI／Oポートピンのマルチ入力ウェイクアップと割り込みを禁止します。
このレジスタのビットを"0"にセットすることで対応するI／Oポートピンのマルチ入力ウェイクアップと割り込みを許可します。

3．5．2．6 WKED_B- ウェイクアップエッジ選択レジスタ

7	6	5	4	3	2	1	0
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

このレジスタのビットを"1"にセットすることで対応するI／Oポートピンのエッジ検出を立ち下がりに選択します。
このレジスタのビットを"0"にセットすることで対応するI／Oポートピンのエッジ検出を立ち上がりに選択します。

3．5．2．7 CMP_B- コンパレータ許可レジスタ

7	6	5	4	3	2	1	0
CMP_EN_	CMP_OE_	res					CMP_RES

CMP_EN_ - アクティブロウ - コンパレータ許可
CMP_OE_ - アクティブロウ - コンパレータ出力許可
res - 予約 -
CMP_RES - comparator result (CMP_EN_ = 0, CMP_OE_ = X)

3．5．3 ポートCレジスタ

ポートCのI／Oピンを構成するために使われる4つのレジスタがあります。TRIS_Cレジスタは、入力か出力かのデータ方向を設定します。LEVL_Cレジスタは、入力ピンをTTLあるいはCMOS電圧レベルに設定します。
PLP_Cレジスタは、ポートCピンのプルアップ抵抗を許可／禁止します。ST_Cレジスタは、ポートC入力ピンのシュミットトリガー入力を許可／禁止します。これらのレジスタにアクセスするためには、MODEレジスタに指定された値を最初に書き込まなければなりません。
ポートCにアクセスするために設定するMODEレジスタの値はTable 3.3を参照して下さい。これらのレジスタのビットは、パワーアップ時に"1"にセットされます。

3．5．3．1 RC- データ方向レジスタ

7	6	5	4	3	2	1	0
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0

このレジスタのビットを"1"にセットすることで対応するI／Oポートピンを入力モード（ハイインピーダンス）にセットします。
このレジスタのビットを"0"にセットすることで対応するI／Oポートピンを出力モードにセットします。

3．5．3．2 LVL_C-TTL/CMOS選択レジスタ

7	6	5	4	3	2	1	0
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0

このレジスタのビットを"1"にセットすることで対応するI／Oポートピンの入力レベルをＴＴＬにセットします。
このレジスタのビットを"0"にセットすることで対応するI／Oポートピンの入力レベルをＣＭＯＳにセットします。

3．5．3．3 PLP_C- プルアップ抵抗許可レジスタ

7	6	5	4	3	2	1	0
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0

このレジスタのビットを"0"にセットすることで対応するI／Oポートピンのプルアップ抵抗を許可します。
このレジスタのビットを"1"にセットすることで対応するI／Oポートピンのプルアップ抵抗を禁止します。

3．5．3．4 ST_C- シュミットトリガー許可レジスタ

7	6	5	4	3	2	1	0
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0

このレジスタのビットを"0"にセットすることで対応するI／Oポートピンのシュミットトリガー入力を許可します。
このレジスタのビットを"1"にセットすることで対応するI／Oポートピンのシュミットトリガー入力を禁止します。

3．5．4 MODEレジスタとポートコントロールレジスタの設定

7	6	5	4	3	2	1	0
0	0	0	0	M3	M2	M1	M0

M3:M0 TRISモードビット - MODE命令、MOV !M,W あるいは MOV W,!M 命令を使って読み書きを行います。

ポートの設定は、MODEレジスタに適切な値を書き出すこと、適切な値をWレジスタにロードすること、または、設定するポートに対してTRIS命令を実行することで行うことが出来ます。以下は、いくつかの例です。

```
mov W,#0Fh
mov !M,W        ;Set up MODE for Data
                ;Direction configuration
mov !RA,#03h    ;RA3:RA2 = Outputs,
                ;RA1:RA0 = Inputs
mov W,#0Eh
mov !M,W        ;Set up MODE for Pull-Up
                ;configuration
mov !RA,#01h    ;RA3:RA1 = Unaffected,
                ;RA0 = Pull-up enabled
mov W,#0Dh
mov !M,W        ;Set up MODE for
                ;TTL/CMOS configuration
mov !RA, #02h   ;RA3,RA2,RA0 = Unaffected,
                ;RA1 = CMOS
```

Table 3.3: MODEレジスタ割り当て

モード	ポートA	ポートB	ポートC
0Fh	WからTRIS_Aにコピー。	WからTRIS_Bにコピー。	WからTRIS_Cにコピー。
0Eh	WからPLP_Aにコピー。	WからPLP_Bにコピー。	WからPLP_Cにコピー。
0Dh	WからLVL_Aにコピー。	WからLVL_Bにコピー。	WからLVL_Cにコピー。
0Ch		WからST_Bにコピー。	WからST_Cにコピー。
0Bh		WからWKEN_Bにコピー。	
0Ah		WからWKED_Bにコピー。	
09h		WとWKEN_Bを交換。	
08h		WとCOMP_Bを交換。	
00h - 07h			

3．5．5 オプション

7	6	5	4	3	2	1	0
RTW	RTE_IE	RTS	RTE_ES	PSA	PS2	PS1	PS0

OPTIONX = 0の時、ビット7と6は、実行されます。
OPTIONX = 1の時、ビット7と6は、"1"として読み出されます。
RTW - 0の時、01h レジスタの読み出しは、Wを結果として得られます。
　　1の時、01h レジスタの読み出しは、RTCCを結果として得られます。
RTE_IE - 0の時、RTCC ロールオーバーは許可されます。
　　1の時、RTCC ロールオーバーは禁止されます。
RTS - 0の時、RTCCは、内部命令サイクルでインクリメントします。
　　1の時、RTCCは、RTCC ピンの変化でインクリメントします。
RTE_ES - 0の時、RTCCは、ロウーハイの変化でインクリメントします。
　　1の時、RTCCは、ハイーロウの変化でインクリメントします。
PSA - 0の時、プリスケーラは、RTCCに割り当てられる。PS0-PS2 ビットで決定される分割の割合によります。
　　1の時、プリスケーラは、WDTに割り当てられる。RTCCの分割の割合は、1:1です。

PS2, PS1, PS0	RTCC 分割の割合	Watchdog Timer 分割の割合
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

3．6 パワーダウンモード

SXはパワーダウンモードを持ち、SLEEP命令を実行することによってパワーダウンモードへ移行します。
ウォッチドッグタイマからの割り込み、割り込み許可されたポートRBピンの外部割込みあるいは、MCLR ピンの外部リセット入力によってパワーダウンモードから抜けることができます。
ウェイクアップ機能は、マルチ入力ウェイクアップ(MIWU)と呼ばれるRBピンを通して動作します。
MIWUによる割り込みから戻るために、SXには2つの命令(RETI と RETI)が用意されています。
パワーダウンモードでは、WDTのみ使用可能です。(WDTが許可されている場合)
SXはSLEEP命令が実行される前のレジスタファイルの内容(データメモリSRAM)と I/Oピンの状態を維持します。
WDTが許可されている場合、SLEEP命令の実行時に、WDTはクリアされ、STATUS レジスタのPDビットがクリア("0")されている間、TOビットはセット("1")されます。
パワーダウンモードにおいてはより消費電力を抑えるため、WDT を禁止し、MIWU または外部デバイスリセットによってパワーダウンモードより抜け出すことをお奨めします。

3．7 割り込みサポート

SXは、内部／外部両方の割り込みをサポートします。外部割込みは、RBポートより行われます。またパワーダウンモードから抜けるためにも使われます。(3．6参照)
内部割込みは、WDTあるいはRTCCどちらかによって発生します。
ＳＸにおける割り込みはすべて同レベルであり，優先順位はありません。
SX は、順番に割り込み処理を行います。割り込み処理を始めると、全体の割り込みは現在行われている割り込み処理から戻るまで禁止されます。
PCは、スタックにプッシュされて、FSR, STATUS, Wの各レジスタの内容は保存されます。割り込み処理から戻ると、PCや上記レジスタの内容は元に戻されます。
割り込みサービスルーチンのベクタアドレスは0です。そして、MIWU、RTCC 及び STATUS レジスタを監視することで割り込みサービスは実行されます。

4．0 メモリ構成

4．1 プログラムメモリ

ＳＸのプログラムメモリ，2Kワードは、4つの512ワードページに分割されています。
命令の順序は、"in-line"命令を実行する毎に自動的にインクリメントするプログラムカウンタ(PC)によってコントロールされています。
分岐を伴う命令の場合は、実行の前にPA1とPA0をセットしなければなりません。
この理由は、"flatten the architecture"によってプログラミングを容易にするために新しくPAGE命令を加えたからです。
8レベルスタック(11ビット幅)により、8レベルの深さまでサブルーチンをネストすることが容易になりました。PAGEの扱い方及び例は、セクション3.4.3を参照して下さい。

4．2 データメモリ

136RAMレジスタは、16レジスタの8バンク分とバンクされていない8レジスタによって構成されています。
バンクと非バンクメモリの両方は、直接的あるいは、FSRを使って間接的にアドレスすることができます。
スペシャルファンクションレジスタは、データメモリにマップされ、読み出しも書き出しもできます。3.4にて、スペシャルファンクションレジスタについて説明しています。
バンクされたメモリの扱い方の説明と例は、セクション3.4.6とFigure 3.2 (FSR)を参照して下さい。

4．3 プログラムメモリのプログラミング

SX マイクロコントローラは、In-systemでプログラムできるように設計されています。SX-KEY 開発ツールは、SX がサーキットボードに取り付けられた後でも、4 ピンのインターフェイスを使って、ターゲットシステムのSXをプログラムすることが出来ます。ターゲットシステムには、4ピンヘッダ以外に必要なものはありません。SX-KEYの電源とGND接続に加えて、OSC1 と OSC2 の2ピンだけがプログラミング中にSXとやり取りするために使われます。OSC1 は、FLASH プログラミングの電源(Vpp=12.5V)を供給するために使われます。OSC2 は、プログラムメモリに対して、データの読み書き、及びデータを正しく書き込めたかを確認するためのシリアルデータ I／Oです。開発が完了した後、内部プログラムメモリへのアクセスを防ぐため、FUSE ワードレジスタには保護ビットが用意されています。

5．0 周辺機器

5．1 オシレータドライバ

5．1．1 概要

SX マイクロコントローラのオシレータドライバは、外部 RC 発振器(RC)，低電圧水晶振動子 (LP)，標準の水晶振動子(XT)，高速水晶振動子 (HS)のいづれかと組み合わせて、システムクロックを発生するために使われます。

5．1．2 機能解説

SXは、5つの異なるオシレータモードで作動させることが出来ます。

.LP: 低電圧水晶振動子
.XT: 標準の水晶振動子
.HS: 高速水晶振動子
.RC: 内部／外部RC発振器

5．1．3 XT, LP あるいは HS モード

XT, LP あるいは HS モードでは、安定に発振させるために水晶あるいはセラミック振動子を OSC1/CLKIN と OSC2/CLKOUT ピンに接続することができます(Figure 5-1)。SX のオシレータドライバデザインには、平行カットタイプの水晶を使用して下さい。連続カットタイプの水晶は、使用しないで下さい。XT, LP あるいは HS モードの時、デバイスは OSC1/CLKIN (Figure 5-2)に外部クロックソースを持つことができ、この場合 OSC2/CLKOUT ピンは、オープンにします。

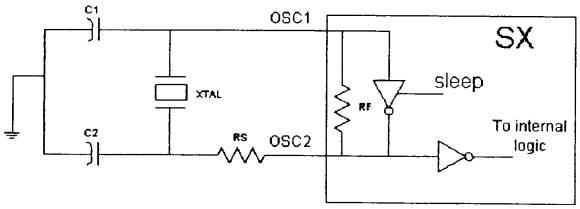


Figure 5.1: 水晶使用 (あるいはセラミック振動子) (HS, XT あるいは LP OSC 構成)

Note 1: C1 と C2 の推奨値はコンデンサー選択テーブルを参照して下さい。
Note 2: ＡＴストリップカット水晶を使用する場合、シリーズ抵抗(RS)が必要になります。
Note 3: RF の値は、水晶の種類によって変動します。(およそ 10MΩ)

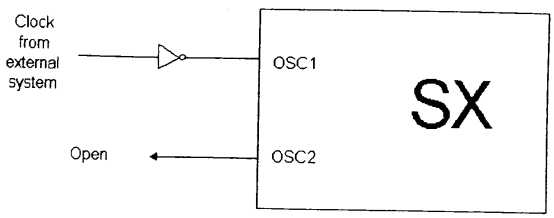


Figure 5.2: 外部クロック入力使用 (HS, XT あるいは LP OSC 構成)

Table 5-1: セラミック振動子のコンデンサー選択

Osc Type	Resonator Freq.	Cap. Range C1	Cap. Range C2
XT	455kHz	68-100pF	68-100pF
	2.0MHz	15-33pF	15-33pF
	4.0MHz	10-22pF	10-22pF
HS	8.0MHz	10-22pF	10-22pF
	16.0MHz	10pF	10pF

Note: コンデンサー C1 は、OSC1 ピンに接続され、コンデンサー C2 は、10MΩ抵抗を経て OSC2 ピンに接続されます。

Table 5-2: 水晶発振のコンデンサー選択

Osc Type	Resonator Freq.	Cap. Range C1	Cap. Range C2
LP	32kHz	15pF	15pF
XT	100kHz	15-30pF	200-300pF
	200kHz	15-30pF	100-200pF
	455kHz	15-30pF	15-100pF
	1.0MHz	15-30pF	15-30pF
	2.0MHz	15pF	15pF
	4.0MHz	15pF	15pF
HS	4.0MHz	15pF	15pF
	8.0MHz	15pF	15pF
	20.0MHz	15pF	15pF
	50.0MHz	15pF	15pF

Note 1: Vdd が 4.5V 以上では、C1 = C2 = 30pF を推奨します。

5．1．4 外部RCモード

タイミングがあまり重要でないアプリケーションには、RC デバイスオプションを使用することによってコストダウンが可能です。RC 発振周波数は、供給電圧、抵抗(Rext)とコンデンサー(Cext)の値と動作温度で決まります。これに加えて、発振周波数は、通常のパラメータの変動により個々に変化します。なおその上に、パッケージ間のリードフレームの容量の違いが、とりわけ低い Cext 値が、発振周波数に影響します。よって、使用する外部の R と C の成分による変動を考慮する必要があります。
Figure 5-3 は、SX に接続される RC の組み合わせ方法を示します。Rext への値は 2.2kΩ以下では、発振動作は不安定となるか、あるいは完全に止まるかもしれません。とても高い Rext 値では(例えば 1MΩ)、発振器は、ノイズ、湿度や漏電にとても敏感になります。従って、Rext は 3kΩから 100kΩに保つことを推奨します。
たとえ外部コンデンサーなし(Cext = 0pF)で使用するとしても、安定性とノイズの理由で 20pF 以上の値を使うことを推奨します。なしあるいは小さいコンデンサーでは、発振周波数は、外部容量の変化により、例えば PCB パターン容量あるいはパッケージリードフレーム容量で劇的に変動します。発振周波数(4 で割った)は、OSC2/CLKOUT ピンで利用可能でテスト目的あるいは他のロジックとの同期に使用できます。

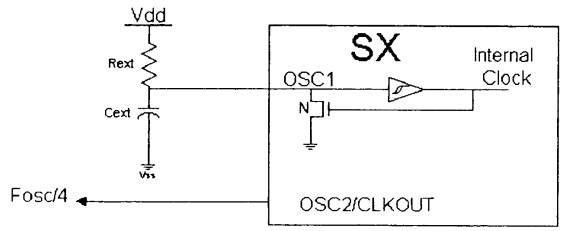


Figure 5.3: RC 発振モード

5．1．5 内部RCモード

内部 RC 発振は、いろいろなアプリケーションに、クロックを供給するための外部回路なしでシングルチップソリューションを提供します。内部 RC の仕様は、電気特性(セクション 6)を参照して下さい。分割できるクロックは、低消費電力での 8 つの低周波数モードを提供しています。範囲は、31.25 KHz から 4MHz までです。
セクション 3.3.1 の FUSE レジスタの DIV2:DIV0 を参照して下さい。

5．1．6 I／O解説

5．1．6．1 入力

OSC1/CLKIN - 水晶発振、RC 入力、外部クロックソース入力

LP_sel - LP mode.
XT_sel - XT mode.
HS_sel - HS mode.
RC_sel - RC mode.
Sleep - スリープモード。発振ドライバは OFF になります。

5．1．6．2 出力

OSC2/CLKOUT - 水晶発振出力
水晶発振モードで、水晶あるいは振動子を接続します。内部および外部 RC モードでは、OSC2 ピンは、内部プルアップ抵抗でプルアップされます。
Clk - システムクロック

5．2 RTCC - (01H) リアルタイムクロックとオプションレジスタ

RTCC は、各命令サイクル毎あるいは RTCC ピンの入力サイクルによりインクリメントする 8 ビットリアルタイムタイマです。内部プリスケーラの設定により RTCC を 16 ビット幅まで拡張することができます。プリスケーラはウォッチドッグタイマあるいは RTCC に割り当てることができますが、両方同時はできませんので注意して下さい。

RTCC モジュールのインクリメントソース

各命令サイクルで RTCC をインクリメントするためにはオプションレジスタ bit5 の RTS ビットをクリアします。このクロックモードでは、RTCC はあらゆる命令サイクル毎(プリスケーラ無し)にインクリメントされます。
RTCC ピンの入力サイクルで RTCC をインクリメントするためには、オプションレジスタ bit5 の RTS ビットをセットします。このカウンタモードでは、RTCC は RTCC ピンの入力サイクル毎にインクリメントされます。

カウンタモードエッジ選択

RTE ビット(OPTION.4)で立ち上がりあるいは立ち下がりのどちらの RTCC 信号のエッジかを決定することが出来ます。実際に RTCC レジスタをインクリメントします。RTE がセットされる時、RTCC は、RTCC ピンの立ち下がりエッジでインクリメントします。逆に RTE がクリアされる時、RTCC は、RTCC ピンの立ち上がりでインクリメントします。

プリスケーラ

8 ビットのプリスケーラは、RTCC あるいは WDT どちらかに割り当てることが出来ます。割り当ては、PSA ビット(OPTION.3)の設定によって行うことが出来ます。PSA ビットがセットされる時、プリスケーラは、WDT タイマに割り当てられ、TMR0 の分割比は 1:1 です。PSA ビットがクリアされる時、プリスケーラは、TMR0 に割り当てられ、分割比は OPTION レジスタにある PS2, PS1, PS0 ビットの状態で決定されます。プリスケーラは、データメモリにマップされず、ランタイムアクセスは出来ません。

OPTION レジスタ

OPTION は、プリスケーラと RTCC の構成に使われるコントロールビットを含むライトオンリーレジスタです。ターボモードでは、さらに RTW と RTE の 2 つのビットが有効になります。

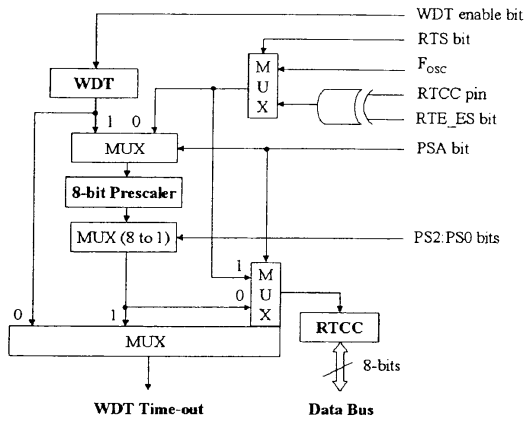


Figure 5.4 RTCC と WDT 回路

5．3 ウォッチドッグタイマ

ウォッチドッグタイマは、RTCC と同じくスケーラを共有しています。(Figure5.4 を参照)
スケーラは、WDT のポストスケーラとして使われます。また、RTCC のプリスケーラとしても使われています。

5．4 コンパレータ

SX は、内蔵差分コンパレータを提供します。ポートピン RB0-RB2 は、コンパレータ出力、－入力、＋入力として個々に使われます。RB ポートピンを通じてコンパレータ I／O にアクセスすることに加えて、コンパレータの操作は、PORTB_CMP 構成レジスタを通じて制御されます。このレジスタは、コンパレータの許可／禁止の制御、内部／外部でのコンパレータ出力の読み出し、コンパレータ出力の許可／禁止等々の設定を行うことが出来ます。詳細は、セクション 3.5.2.7 を参照して下さい。コンパレータの選択／制御ビットは、RESET でクリアされ、コンパレータは禁止されます。低電圧消費を必要とするアプリケーションでは、コンパレータはパワーダウンモードに入る前に禁止にすることを推奨します。正しい操作としては、ユーザーは、コンパレータの入力／出力として正しく RB0, RB1 と RB2 ポートをプログラムしなければなりません。言い換えれば、RB1 と RB2 は、入力としてセットアップしなければならず、RB0 は、出力としてセットアップしなければなりません。コンパレータの AC/DC 仕様は、セクション 6 を参照して下さい。

5．5 リセット

5．5．1 概要

パワーアップとブラウンアウト間でリセットを発生するために SX マイクロコントローラで使われる RESET 回路について説明します。

5．5．2 機能解説

SX には、パワーアップ時の内部チップリセットを提供するパワーオンリセット(POR)回路が組み込まれています。電圧(VDD)がその最小値以下に降下する時(0 ではない)、ブラウンアウト回路は、チップをリセットして、回復させます。

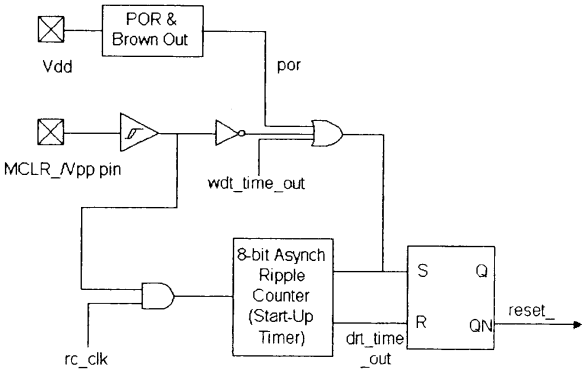


Figure 5.5: 内蔵リセット回路のブロックダイヤグラム

パワーオンで、リセットラッチはセットされ、ディレイリセットタイマ(DRT)がリセットします。DRT タイマは、MCLR_がハイになることを検出するとすぐ、カウントを始めます。72ms であるタイムアウト期間後、ラッチリセットはリセットします。このようにして内蔵リセット信号は終了します。
VDD に無関係の MCLR_のパワーアップ例は、Figure 5.6 で示します。VDD は立ち上がりを許容され、MCLR_がハイになる前に安定します。チップは、現実には MCLR_がハイになる後、Tdrt ms でリセットから抜けます。

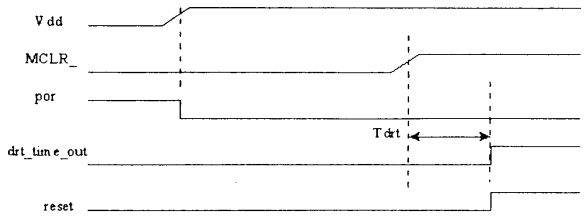


Figure 5.6 パワーアップのタイムアウトシーケンス (MCLR_は VDD に無関係)

Figure 5.7 では、内蔵パワーオンリセットの特長は使われています(MCLR_と VDD は、相互関係あり)。VDD は、スタートアップタイマがタイムアウトする前に安定して、正常なリセット動作に問題はありません。しかしながら、VDD が遅く立ち上がりの問題の状態では Figure 5.8 のようになります。DRT が MCLR_/VPP ピンをハイとする時と MCLR_/VPP ピン(それと VDD)が現実に最大値に達する時の間の時間が長すぎるのです。この状態では、スタートアップタイマがタイムアウトする時、VDD は、VDD(最小)値に達せず、従ってチップは、正しい動作が保証されません。そのような状態には、長い POR ディレイタイムを成すために使われる外部 RC 回路を推奨します(Figure 5.9)。

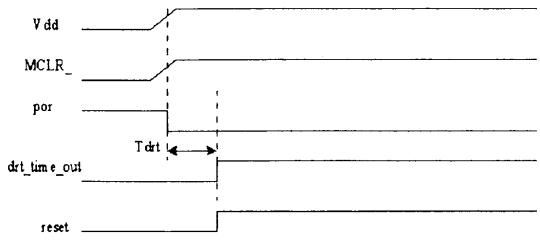


Figure 5.7 パワーアップ時のタイムアウトシーケンス
(MCLR_は VDD に関係):早い VDD の立ち上がり時間

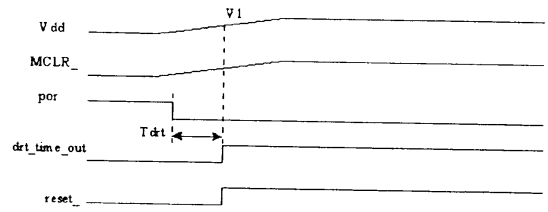


Figure 5.8 パワーアップ時のタイムアウトシーケンス
(MCLR_は VDD に関係):遅い立ち上がり時間

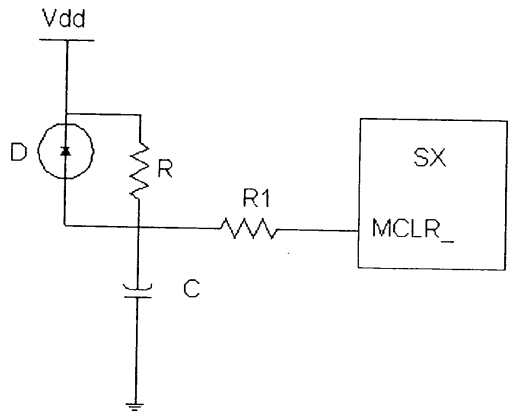


Figure 5.9 外部パワーオンリセット回路 (パワーアップが遅い VDD 用)

Note 1: VDD パワーアップがあまりに遅い場合にのみ、外部パワーアップリセット回路が要求されます。VDD のパワーが落ちる時、ダイオード D は、早くコンデンサーを放電させます。
Note 2: R < 40 kΩは、R を越えた電圧降下がデバイスの電気特性に違反しないことを確認することが求められます。
Note 3: R1 = 100Ω to 1kΩは、静電気(ESD)あるいは過電流による MCLR_ピン故障の出来事での外部コンデンサー C から MCLR_への電流流出を制限します。

5．6 ブラウンアウト検出

ブラウンアウトは、デバイス電圧(VDD)がその最小値以下に降下する時(0 ではない)、回復する状態をいいます。デバイスは、ブラウンアウトのイベントでリセットしなければなりません。ブラウンアウトレベルは、4V 未満より少し前にプリセットされます。また、ブラウンアウトは、brown-out 信号によってもキャンセルされます。ブラウンアウト回路が切られる時、チップの状態はちょうど POR になるところです。

6．0 電気特性 (暫定仕様)

6．1 絶対最大定格 (暫定仕様)

動作周囲温度	-40℃ to +85℃
保存温度	-65℃ to +150℃
Voltage on Vdd with respect to Vss	0V to +7.5V
Voltage on MCLR with respect to Vss	0V to +14V
Voltage on all other pins respect to Vss	0.6V to (VDD + 0.6V)V
総消費電力	800mW
Max. current out of Vss pin	150mW
Max. current into Vdd pin	100mW
Max. current into an input pin (TOCKI only)	±500mA
入力クランプ電流, Iik (Vi<0 or Vi>Vdd)	±20mA
出力クランプ電流, Iok (Vo<0 or Vo>Vdd)	±20mA
最大サンク電流 by any I/O pin	30mA
最大ソース電流 by any I/O pin	30mA
最大ソース電流(1端子/ポートAあるいはB)	TBD
最大サンク電流(1端子/ポートAあるいはB)	TBD

6．2 DC 特性 (暫定仕様)

動作温度 0℃ <= Ta <= +70℃ (Commercial)

略号	項目	Mode	Min	Typ	Max	単位
Vdd	電源電圧	RC	3.3	-	6.25	V
		XT	3.3		6.25	
		HS	3.3		6.25	
		LP	3.3		6.25	
Vpor	POR動作保証最小Vdd電	-	Vss	-	V	
Svdd	Vdd立ち上がり速度		0.05	-	-	V/ms

電源電流

略号	項目	Min	Typ	Max	単位	条件
Idd	電源電流(動作時)	-	40	-	mA	Vdd=5V, Fosc=50MHz
			1.5		mA	Vdd=5V, Fosc=4MHz内蔵RC
			15		mA	Vdd=3.3V, Fosc=20MHz
Isd	電源電流(スリープ時)	-	TBD	-	μA	Vdd=3.3V, WDT許可時
			TBD		μA	Vdd=3.3V, WDT禁止時
Ipd	電源電流(スタンバイ時	-	TBD	-	μA	Vdd=3.3V, WDT許可時
			10		μA	Vdd=3.3V, WDT禁止時

6．3 AC 特性 (暫定仕様)

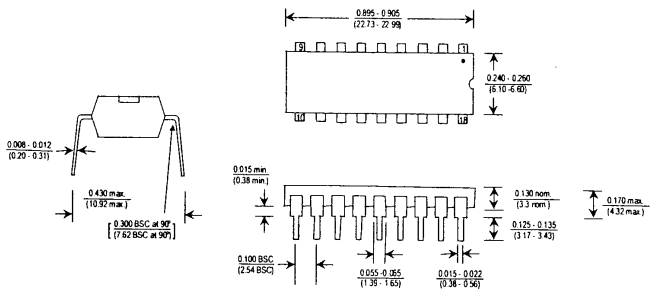
略号	項目	Min	Typ	Max	単位	条件
Fosc	外部CLKIN周波数	DC	-	4.0	MHz	RC
				4.0	MHz	XT
				50	MHz	HS
				200	Khz	LP
	発振周波数	DC	-	4.0	MHz	RC
		0.1		4.0	MHz	XT
		4		50	MHz	HS
		5		200	Khz	LP
Tosc	外部CLKIN周期	250	-	-	nS	RC
		250		10,000	nS	XT
		20		250	nS	HS
		5.0		200	μS	LP
TosL, TosH	クロック入力(OSC1) Low/High幅	50	-	-	nS	XT
		8.0			nS	HS
		2.0			μS	LP
TosR, TosF	クロック入力(OSC1) Rise/Fall時間	-	-	25	nS	XT
				25	nS	HS
				50	μS	LP

Note: "TYP"項目の条件は、5V, 25℃です。

6．4 コンパレータ DC/AC 仕様 (暫定仕様)

項目	条件	Min	Typ	Max	単位
入力オフセット電圧	0.4V<Vin<Vcc - 1.5V		±10	±25	mV
入力コモン電圧レンジ		0.4		Vcc - 1.5	V
電圧ゲイン			300k		V/V
DC電源電流 (動作時)	Vcc = 5.5V			250	μA
応答時間				1	μS

SX18AC/DP



IN-SYSTEM PROGRAMMING

INTRODUCTION

In-system-programming is the capability to program or re-program a chip in place after it has been mounted on a circuit board. The benefits of in-system programming have been well documented. It speeds up product development, minimizes the risk of over stocking of out-dated parts, and allows field upgrade and bug fix.

With ISP, developing a micro-controller system is simpler and faster. During program development, codes can be updated with the micro-controller in the actual system board. Without ISP, the micro-controller has to be removed from the board and plugged into a separate programmer before it can be programmed. The controller can be program and re-program many times, using in-system-programming feature, until the design functions correctly. It saves a lot of time. Also, the micro-controller can be soldered on the board just like the actual production system. No special sockets are needed. It makes debugging timing or noise sensitive designs a lot easier.

For manufacturing, there are many benefits too. System boards can be built with the controller mounted, without sockets, before the design is finalized to meet time-to-market requirement. Additional information can be added at end of production cycles, information that may not be available otherwise, such as, vendor ID. But, one of the major benefits is to reduce the risk of over stocking out-dated pre-programmed devices. Because the controller can be programmed or re-programmed while in system, the parts do not have to be pre-programmed. And pre-programmed parts can be re-programmed if revisions or upgrades become necessary. Finally, with ISP, the parts can be easily upgraded or fixed in the field.

The ISP solution from Scenix is a proprietary system but uses the two pins, clock input (OSC1) and clock output (OSC2), for crystal oscillator. It does not require an expensive JTAG tester and, in most applications, does not need any additional hardware to isolate the ISP circuitry from other hardware on the system board. The reason is that OSC1 and OSC2 are normally connected to passive components such as capacitors, resistors, or crystals and can not be damaged with high programming voltages and would not interfere with ISP programming signals. The Scenix solution does not use an expensive tester, does not add pins to the micro-controller and does not need extra hardware. It reduces component cost and real estate requirement on the circuit board.

In-System-Programming

The in-system-programming (ISP) mechanism allows SX to be programmed in the system board without removing the SX. In most cases, no modifications to the system board are needed. Only two pins, OSC1 and OSC2 are used. OSC1 is used to supply power (Vpp = 12.5V) for EEPROM programming. OSC2 is the serial data I/O for writing data to EEPROM and reading data from EEPROM.

There are three stages of in-system-programming: Entering ISP, Programming ISP, and Exiting ISP.

Entering ISP:

The method to enter ISP depends on the oscillator mode selected. SX has three oscillator modes: crystal (XTAL), external RC (XRC), and internal RC (IRC). The three methods are slightly different but the basic idea is the same: signal SX to shut off the oscillator at OSC1 so that a high voltage can be applied at OSC1 for EEPROM programming (Fig. 1).

OSC2 is the communication pin for ISP. To tell SX to shut off the oscillator, drive OSC2 low. To avoid the oscillator from being inadvertently shut off, a noise filtering logic is used. OSC2 has to be low for nine consecutive OSC1 clocks to signal the beginning of ISP.

Use the following steps to start ISP:

First, force OSC1 low. Then force OSC2 low. With OSC2 low, toggle OSC1 nine times. These nine OSC1 clocks, driven by the ISP programming module (e.g. ISP programming module from Parallax, Inc.), clears OSC_EN bit in the DEBUG register and disables the on chip oscillator circuit.

With the oscillator circuit disabled, OSC1 can be safely driven to Vpp level without any contention. The last step is to raise OSC1 to VPP. When VPP appears at OSC1, the internal RC oscillator (freq. = 128K) is enabled and becomes the clock for ISP state machine. The ISP logic is now ready for programming.

Programming ISP:

When ISP programming stage is entered, SX could be in the middle of executing a program. To prevent any accidental damage to the system board, the first thing the ISP logic does is to reset the chip. When reset is done, ISP logic executes the in-system-program protocol. It is a "self aligned" protocol. One pin, OSC2, is used for both serial data I/O and a "synchronization clock". No separate clock pin is needed. OSC2 is implemented as open drain with an internal pull up.

The ISP programming protocol is made up of a stream of ISP frames at OSC2. In the current scheme, a frame has 17 cycles and a cycle has four clocks (Fig.2).

The first cycle of a frame is the synchronization cycle. It signals the beginning of a frame. It is followed by four cycles for entering ISP command. These are the command cycles. The command is shifted in serially and it specifies an ISP operation. For example, write data to EEPROM or read data from EEPROM. The command phase is followed by twelve cycles for data. These are the data cycles. Depending on whether the ISP command is to write data or read data, EEPROM data is either serially driven onto OSC2 by the external programming module (write case) or by the chip (read case). In either case, data is valid on the raising edge of fourth clock (clock 3).

There are four clocks in a cycle. On the first clock (clock0), nobody drives OSC2. Hence, the open drain OSC2 is pulled high by the internal pull up of OSC2. On second clock (clock 1), the chip drives OSC2 low. This is the synchronization pulse. The ISP programming module uses this pulse and the synchronization cycle to align data for writing to or reading from the chip. The third and fourth clocks (clock2 and clock3) are for data. Depending on the direction of data transfer, the external module or the chip can drive data during these two clocks. The data is valid and sampled on the rising edge of clock3.

One special note on the operation is that the encoding for the "no operation" command, NOP, must be all one's. This is important because once entering ISP programming stage, the ISP logic on chip will start looking for command to execute. But, the external ISP programming module takes some time to align its internal clock to the synchronization cycle and synchronization pulses so that command can be driven onto OSC2 at the right cycles (command cycles) and at the right clocks (clock2 and clock3). Since OSC2 is open drain, ISP programming module, by not driving OSC2 (and thus no synchronization is needed), effectively drives the NOP command. When the ISP programming module aligned its internal clock with the synchronization pulses, it can safely and correctly drive the command and write or read data.

Currently, nine commands are defined:

Name	Code	Description
NOP	0xf	No operation.
ERASE	0x0	Erase ALL EEPROM locations.
READ DEV	0x1	Read SX features available for the device.
READ FuseX	0x2	Read FuseX.
PROG FuseX	0x3	Write FuseX. Need to execute the LOAD command first to specify the data to be programmed.
LOAD	0x4	Specify the data to be programmed.
PROG	0x5	Write data to EEPROM other that FuseX. Need to execute the LOAD command first to specify the data to be programmed.
READ	0x6	Read EEPROM data other than FuseX.
INC	0x7	Increment EEPROM address by one.

However, the scheme does not put any restriction on the number of commands. New commands can be added as needed.

The on chip ISP logic is consisted of three major blocks: command shift register, command decode and data shift register. Command is shifted in serially and decoded once all command bits are shifted in place. In a write data operation, data to be written is first loaded in the data register by shifting the data in serially, using the LOAD command. It is then written to the EEPROM in parallel using the PROG command. To control the length of data write time, simply execute the PROG command continuously for the desired number of times. In a read data operation, use the READ to load the data shift register with EEPROM data in parallel mode and then to shift the data out serially. For example, to program one location in EEPROM, following commands need to be executed:

1. ERASE (First, erase the memory. Issue the ERASE command several time to allow sufficient time to erase. For example, if the erase time is x msec and each frame takes 0.53msec (1/128k * 4 * 17), issue ERASE command x/0.53 times)
2. LOAD
3. PROG (Issue PROG command several times to allow sufficient time to program. For example, if the program time is x msec and each frame takes 0.53msec (1/128k * 4 * 17), issue PROG command x/0.53 times)

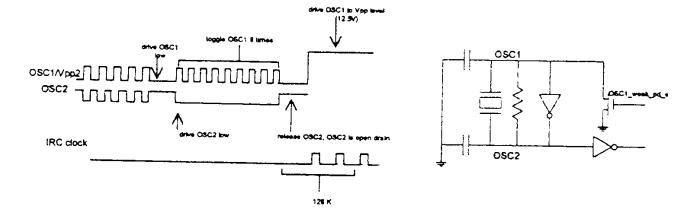
Exiting ISP:

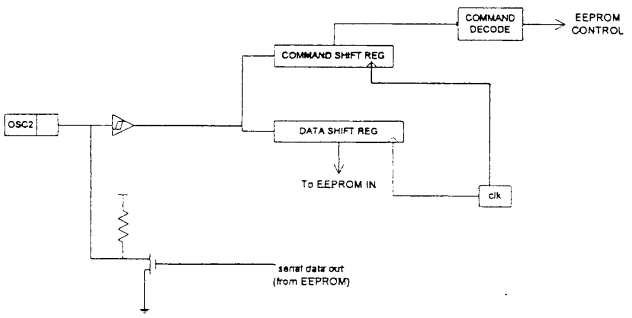
When ISP programming is done, a method is needed to safely exit the ISP mode without accidentally damaging the system board. A scheme is needed to let both the on chip ISP logic and the off chip ISP programming module know that it is safe to exit ISP.

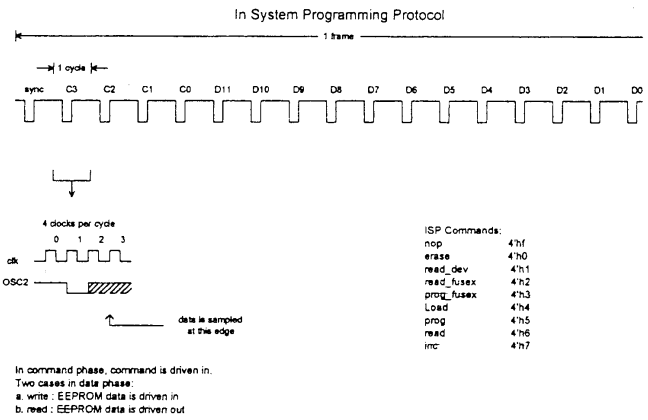
The scheme chosen is for the off chip ISP programming module to drop OSC1 from Vpp to low. This signals its intention to exit ISP mode. The ISP mode is exited on the first rising clock edge after the synchronization cycle. At this time, the on chip ISP logic generates a reset pulse to reset the chip and terminate the ISP mode. Before this point, both on chip and off chip ISP logic need to observe all ISP protocol.

3 OSC modes: XTAL, XRC, IRC

1.) XTAL





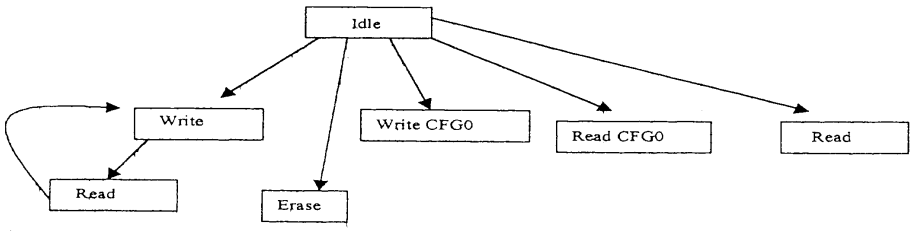


SX PARALLEL MODE PROGRAMMING SPECIFICATION

The SX chip can be programmed in parallel mode also. The pins that are used for parallel mode programming are Vdd, MCLR_/VPP, portB, portA, RTCC and OSC1 pins. To program one location in EEPROM, following commands need to be executed:

1. ERASE (Issue the ERASE command and allow sufficient time to erase the memory.)
2. PROGRAM (Issue PROG command and allow sufficient time to program. The PROGRAM command does WRITE and READ operation.)

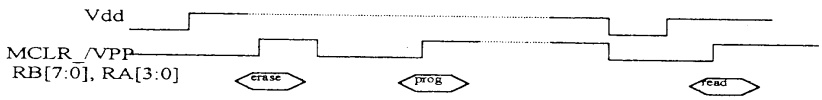
Fig.1 shows the necessary timing diagram of the signals needs to be applied at the pins to program the chip in parallel mode.
The state machine for the parallel programming block is as follows.



Commands:	Hex Code:
Erase	hx1x
Read device info	h001
Read FUSEX	h002
Write FUSEX (SX28)	h003
Write FUSEX (SX18)	hx2x
Read	h004
Program	hFFE

Programming Specification Additional Information:

- Recommend a protection resistor of 100Ω on MCLR_/V_{pp} pin.
  - V_{pp} = 12.5V
- Command is latched at the rising edge of MCLR_/Vpp signal. The rise time of MCLR_/Vpp signal should be greater than 1µs.
- There should be atleast 1µs hold time for the data to be written after the rising edge of RTCC.
- There is a 50ns-delay time for the data to be read after the falling edge of RTCC.
- **For 18 and 20 pin package, after ERASE command, first need to PROGRAM the bit 10 of FUSEX word to 0.**
- Address is incremented on the leading edge transition of OSC1
  - It can be advanced independent of RTCC
  - To skip "X" number of locations, just toggle OSC1 "X" times. Then drive RTCC low to WRITE (if the command is to write) or READ (if the command is to read) the location.
- Operation is specified by command
  To do a series of commands, cycle Vdd, or Vpp, or both.



- SX18 and SX28 use different commands to write to FUSEX word register.