SX Reset Considerations

Application Note 18
Zafar Ullah
September 1999

1.0 Overview

This application note discusses in detail the SX reset conditions. It provides information about the on-chip reset circuitry and suggests software methods to distinguish between the various reset conditions.

Power-On-Reset, Brown-Out reset, watchdog reset, or external reset initializes the device. Each one of these reset conditions causes the program counter to branch to the top of the program memory. For example, on the device with 2048K words of program memory, the program counter is initialized to 07FF.

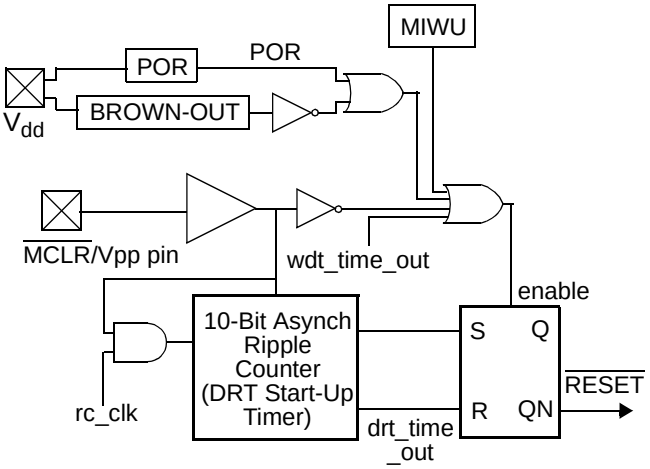
Multi-Input Wakeup is one way of causing the device to exit the power down mode (Sleep Mode). Port B is used to support this feature.Once a valid transition occurs on the selected pin, the WKPND_B register (Wakeup Pending Register) latches the transition in the corresponding bit position. A logic '1' indicates the occurrence of the selected trigger edge on the corresponding Port B pin. Upon exiting the power down mode, the Multi-Input Wakeup logic causes program counter to branch to the maximum program memory address (same as reset).

The on-chip brown-out detection circuitry resets the device when V_{dd} dips below the specified brown-out voltage. The device is held in reset as long as V_{dd} stays below the brown-out voltage. The device will come out of reset when V_{dd} rises above the brown-out voltage.The two bits in the FUSEX word (BOR1:BOR0) selects the level of the brown-out voltage.

The SX device contains an 8-bit Watchdog Timer (WDT). If the prescaler is not used for the RTCC, it can serve as a postscaler for the Watchdog Timer. The Watchdog is designed to detect user program getting stuck in the infinite loops resulting in loss of program control and also used for periodically waking SX up from the sleep mode. The Watchdog can be enable or disable by the WDTE bit in the FUSE Word. On power-up, the Watchdog is disabled. Watchdog timer underflow causes program counter to branch to the maximum program memory address (same as reset)

2.0 On-Chip Reset Circuitry

The device incorporates an on-chip Power-On Reset (POR) circuit that generates an internal reset as V_{dd} rises during power-up. Figure 2-1 is a block diagram of the circuit. The circuit contains an 10-bit Delay Reset Timer (DRT) and a reset latch. The DRT controls the reset timeout delay. The reset latch controls the internal reset signal. Upon power-up, the reset latch is set (device held in reset), and the DRT starts counting once it detects a valid logic high signal at the $\overline{MCLR}$ pin. Once DRT reaches the end of the timeout period (typically 72 msec), the reset latch is cleared, releasing the device from reset state.



Note:Ripple counter is 10 bits for Power on Reset (POR) only.

Figure 2-1. Block Diagram of On-Chip Reset Circuit

Figure 2-2 shows a power-up sequence where $\overline{MCLR}$ is not tied to the V_{dd} pin and V_{dd} signal is allowed to rise and stabilize before $\overline{MCLR}$ pin is brought high. The device will actually come out of reset T_{drt} msec after $\overline{MCLR}$ goes high.

The brown-out circuitry resets the chip when device power (V_{dd}) dips below its minimum allowed value, but not to zero, and then recovers to the normal value.

Figure 2-3 shows the on-chip Power-On Reset sequence where the $\overline{MCLR}$ and V_{dd} pins are tied together. The V_{dd} signal is stable before the DRT time-out period expires. In this case, the device will receive a proper reset. However, Figure 2-4 depicts a situation where V_{dd} rises too slowly. In this scenario, the DRT will time-out prior to V_{dd} reaching a valid operating voltage level (V_{dd} min). This means the device will come out of reset and start operating with the supply voltage not at a valid level. In this situation, it is recommended that you use the external RC circuit shown in Figure 2-5. The RC delay should exceed the time period it takes V_{dd} to reach a valid operating voltage.

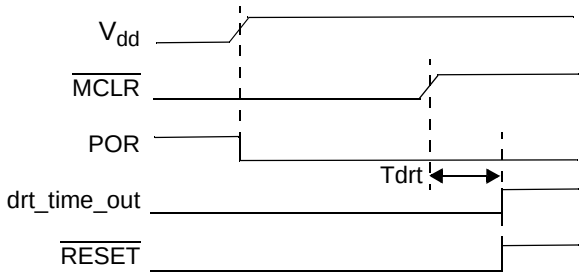


Figure 2-2. Time-Out Sequence on Power-Up (MCLR not tied to V_{dd})

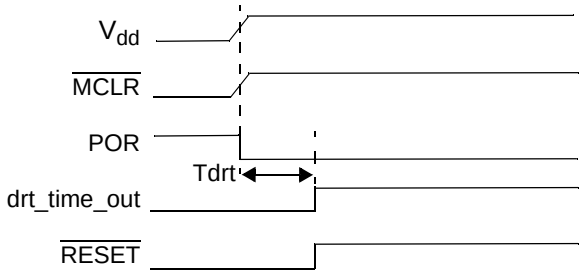


Figure 2-3. Time-out Sequence on Power-up (MCLR tied to V_{dd}): Fast V_{dd} Rise Time

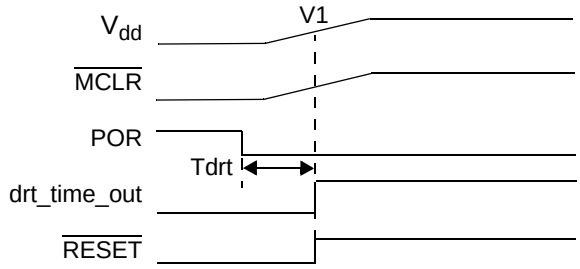


Figure 2-4. Time-out Sequence on Power-up (MCLR tied to V_{dd}): Slow Rise Time

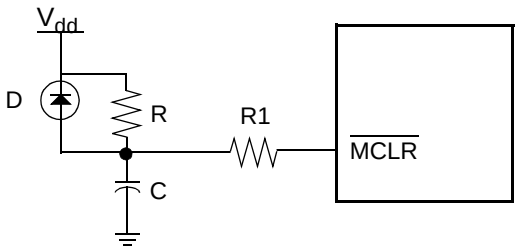


Figure 2-5. External Power-On Reset Circuit (For Slow V_{dd} Power-up)

Note 1: The external Power-On Reset circuit is required only if V_{dd} power-up is too slow. The diode D helps discharge the capacitor quickly when V_{dd} powers down.

Note 2: R < 40 kΩ is recommended to make sure that voltage drop across R does not violate the device electrical specifications.

R1 = 100Ω to 1kΩ will limit any current flowing into $\overline{MCLR}$ from external capacitor C. This helps prevent $\overline{MCLR}$ pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

2.1 DELAY RESET TIMER (DRT)

The Delay Reset Timer (DRT) timeout period is 18.4 msec for the SX18/20/28AC devices.

In SX48/52BD devices, a 3-bit field in the FUSEX register (WDRT2:WDRT0) can be used to specify the Delay Reset Timer (DRT) timeout period that results in an automatic wake-up from the power down mode:

101 = 0.06 msec
110 = 7.68 msec
111 = 18.4 msec (default)
000 = 60 msec
001 = 480 msec
010 = 960 msec
011 = 1920 msec

For fast start-up from the power down mode, clear the SLEEPCLK bit and set the WDRT2:WDRT0 field to 100. This will keep the clock operating during the power down mode and allow a zero start-up delay.

3.0 Register States Upon Different Reset Conditions

The effect of different reset operation on a register depends on the register and the type of reset operation. Some registers are initialized to specific values, some are left unchanged (for wakeup and brown-out resets), and some are initialized to an unknown value. A register that starts with an unknown value should be initialized by the software to a known value; you cannot simply test the initial state and rely on it starting in that state consistently.

Table 3-1 lists the SX18/20/28AC registers and shows the state of each register upon different reset conditions.

Table 3-1. Register States Upon Different Reset Conditions

Register	Power-On	Wakeup	Brown-Out	Watchdog Timeout	MCLR
W	Undefined	Unchanged	Undefined	Unchanged	Unchanged
OPTION	FFh	FFh	FFh	FFh	FFh
MODE	0Fh	0Fh	0Fh	0Fh	0Fh
RTCC (01h)	Undefined	Unchanged	Undefined	Unchanged	Unchanged
PC (02h)	FFh	FFh	FFh	FFh	FFh
STATUS (03h)	Bits 0-2: Undefined Bits 3-4: 11 Bits 5-7: 000	Bits 0-2: Unchanged. Bits 3-4: Unch. Bits 5-7: 000	Bits 0-4: Undefined Bits 5-7: 000	Bits 0-2: Unchnaged Bits 3-4: (Note 1) Bits 5-7: 000	Bits 0-2: Unchanged Bits 3-4: (Note 2) Bits 5-7: 000
FSR (04h)	Undefined	Bits 0-6: Unchanged Bit 7: 1	Bits 0-6: Undefined Bit 7: 1	Bits 0-6: Unchanged Bit 7: 1	Bits 0-6: Unchanged Bit 7: 1
RA/RB/RC Direction	FFh	FFh	FFh	FFh	FFh
RA/RB/RC Data	Undefined	Unchanged	Undefined	Unchanged	Unchanged
Other File Registers - SRAM	Undefined	Unchanged	Undefined	Unchanged	Unchanged
CMP_B	Bits 0, 6-7: 1 Bits 1-5: Undefined	Bits 0, 6-7: 1 Bits 1-5: Undefined	Bits 0, 6-7: 1 Bits 1-5: Undefined	Bits 0, 6-7: 1 Bits 1-5: Undefined	Bits 0, 6-7: 1 Bits 1-5: Undefined
WKPND_B	Undefined	Unchanged	Undefined	Unchanged	Unchanged
WKED_B	FFh	FFh	FFh	FFh	FFh
WKEN_B	FFh	FFh	FFh	FFh	FFh
ST_B/ST_C	FFh	FFh	FFh	FFh	FFh
LVL_A/LVL_B/LVL_C	FFh	FFh	FFh	FFh	FFh
PLP_A/PLP_B/PLP_C	FFh	FFh	FFh	FFh	FFh
Watchdog Counter	Undefined	Unchanged	Undefined	Unchanged	Unchanged

NOTE: 1. Watchdog reset during power down mode: 00 (TO, PD)
Watchdog reset during Active mode: 01 (TO, PD)

NOTE: 2. External reset during power down mode: 10 (TO, PD)
External reset during Active mode: Unchanged (TO, PD)

4.0 Software Considerations

How does the user program determine which reset condition caused the reset?

Based on the rest source, the application may skip part or full of the initialization code, or take some application specific action. One thing is obvious that, whatever the source of the reset, the program counter (PC) becomes initialized to top of the program memory. For example, on the device with 2048K words of program memory, the program counter is initialized to 07FF.

The STATUS register holds the arithmetic status of the ALU, the page select bits, and the reset state. The STATUS register is accessible during run time, except that bits PD and TO are read-only. Table 1-2 shows the status of these two bits (TO, PD) after the reset of the processor.

Table 4-1. Staus of TO, PD Bits After RESET

TO	PD	Reset Source
1	1	After Power Up Reset
1	0	SX was on sleep and reset by Multi-input-wakeup (MIWU)
0	0	SX was on sleep and reset by Watchdog timer (WDT)
0	1	SX was on active mode and reset by WDT
1	0	SX was on sleep and reset by external reset pin (MCLR)
Un-changed	Un-changed	SX was on active mode and reset by external reset pin (MCLR)

It is obvious that these two bits (TO, PD) alone can't determine the source of the reset. But with these two bits and the information from both the tables (Table 1-1 and Table 1-2) will help us to come up with the algorithm to determine the source of the reset.

The Pseudocode and the flowchart describes the software algorithm to determine the source of reset. WKPEND.WKPIN is the bit in the wakeup pending registar which corresponds to the wakeup pin. WKPEND.IND is the bit in the wakeup pending registar which corresponds to the non-wakeup pin.

4.1 FLOW CHART

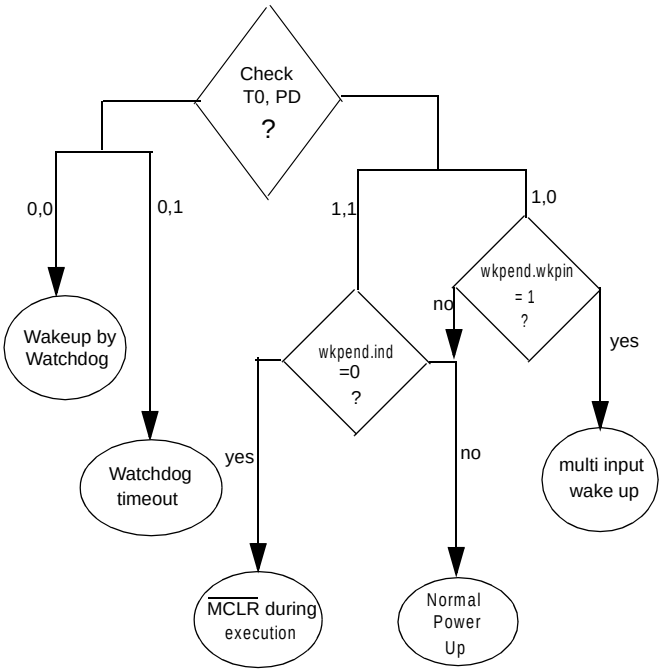


Figure 4-1. Flowchart

4.2 PSEUDOCODE

```
;How_Do_I_Got_Here
; Four reasons could get me here:
; - a transition on the RB.WKPIN
; - a mclr reset while in sleep (power down) mode
; - a mclt reset while running
; - a watchdog wakeup
; - a watchdog timeout
; - or someone turned on power
;
; ALGORYTHM:
 if(TO==1){
        if(PD==0){
                //PD=0 TO=1
        if(WKPEND.WKPIN==1){;rb.wkpin is the miwu pin
                miwu()
        }else{
                power_on()// normal power_up
        }
}else{
        //PD=1 TO=1
        if(WKPEND.IND==0){;rb.IND is a non miwu pin
                mclr_during_exec();
        }else{
                power_on()// normal power_up
        }
}
 }else{
        if(PD==0){
                //PO=0 TO=0
        wakeup_through_watchdog()
}else{
        //PD=1 TO=0
        watchdog_timeout() // clr!wdt more often
}
}
```