Ultra Low Power 8-pin Flash Microcontroller

Features

- **True Low Power:** 5.8 µA active mode
  3.3 µA standby mode
  0.32 µA sleep mode
- **Large Supply Voltage 2.0 V to 5.5 V**
- **No external component needed**
- **Available in SO-8/14 packages and die**
- **4-bit ADC or 12 levels Supply Voltage Level Detector (SVLD)**
- **Unique ID code of 52bits + 16bits CRC**

- Max 4 (5*) outputs with 2 high drive outputs of 10mA
- Max. 5 (6*) inputs
- Sleep Counter Reset (automatic wake-up from sleep mode (EM patent))
- Flash memory 4096 × 16 bits
- RAM 80 × 4 bits
- Internal RC oscillator 32kHz – 800kHz
- 2 clocks per instruction cycle
- 72 basic instructions
- External CPU clock source possible
- Watchdog timer (2 sec)
- Power-On-Reset with Power-Check on start-up
- 3 wire serial port , 8 bit, master and slave mode
- Universal 10-bit counter, PWM, event counter
- Prescaler down to 1 Hz (freq. = 32kHz)
- Frequency output 1Hz, 2048 Hz, CPUClk, PWM
- 6 internal interrupt sources (2×10-bit counter, 2× prescaler, SVLD, Serial Interface)
- 2 external interrupt sources (port A)

Description

The EM6580 is a low power Flash 4-bit microcontroller coming in a small 8-pin SO package and working up to 0.4 MIPS. It comes with an integrated 4-bit ADC and 2 high current drive outputs of 10mA and it requires no external component. It has a sleep counter reset allowing automatic wake-up from sleep mode. It is designed for use in battery-operated and field-powered applications requiring an extended lifetime. A high integration level make it an ideal choice for cost sensitive applications.

The EM6580 contains the equivalent of 8kB of Flash memory and a RC oscillator with frequencies between 32 to 800kHz. It also has a power-on reset, watchdog timer, 10 bit up/down counter, PWM and several clock functions.

Development tools include windows-based simulator program debugger, assembler and real time emulator.

Figure 1. Architecture

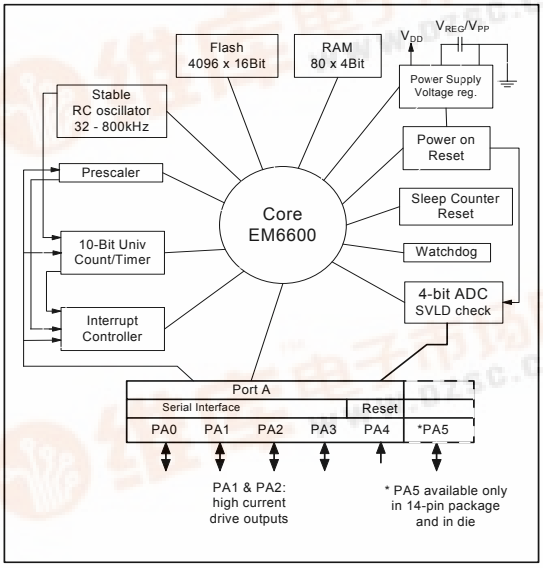
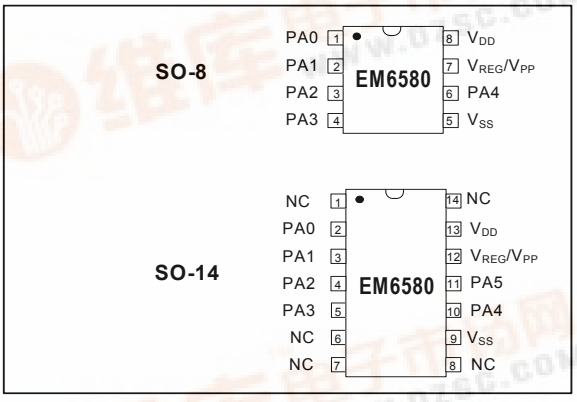


Figure 2. Pin Configuration



Typical Applications

- Household appliances
- Safety and security devices
- Automotive controls
- Sensor interfaces
- Watchdog
- Intelligent ADC
- Driver (LED, triac)

EM6580

EM6580 at a glance

❑ Power Supply
- Low voltage low power architecture including internal voltage regulator
- 2.0V to 5.5V supply voltage
- 5.8 µA in active mode
- 3.3 µA in standby mode
- 0.32 µA in sleep mode

❑ RAM
- 80 x 4 bit, directly addressable

❑ FLASH
- 4096 x 16 bit (8k Byte),

❑ CPU
- 4-bit RISC architecture
- 2 clock cycles per instruction (CPI=2)
- 72 basic instructions

❑ Main Operating Modes and Resets
- Active mode (CPU is running)
- Standby mode (CPU in halt, peripherals running)
- Sleep mode (no clock, data kept)
- Initial Power-On-Reset with Power-Check
- Watchdog reset (logic)
- Reset terminal (software option on PA[3/4])
- Sleep Counter reset from Sleep mode
- Wakeup on change from Sleep mode

❑ Prescaler
- Divider (4 stages) to best fit CPU clock (32kHz – 1MHz to 32kHz system clock to keep peripherals timing close to specification
- 15 stage system clock divider from 32kHz down to 1Hz
- 2 Interrupt requests (3 different frequencies)
- Prescaler reset (4kHz to 1Hz)

❑ 8-Bit Serial Interface
- 3 wire (Clock, DataIn , DataOut) master/slave mode
- READY output during data transfer
- Maximum shift clock is equal to the main system clock
- Interrupt request to the CPU after 8 bit data transfer
- Supports different serial formats
- pins shared with general 4 bit PA[3:0] I/O port

❑ Oscillator
- RC Oscillator range: 32kHz up to 800kHz
- No external components are necessary
- Temperature compensated
- External clock source possible from PA[1]

❑ 4(5)-Bit I/O PA[3:0] & PA[4] / PA[5]*
- Direct input read on the port terminals
- 2 debounced function available muxed on 4 inputs
- 2 Interrupt request on positive or negative edge
- Pull-up or pull-down or none selectable by register
- 2 Test variables (software) for conditional jumps
- PA[1] and PA[3/4] are inputs for the event counter
- PA[3/4] Reset input (register selectable)
- All outputs can be put tri-state (default)
- Selectable pull-downs in input mode
- CMOS or Nch. open drain outputs
- Weak pull-up selectable in Nch. open drain mode

❑ 4-bit ADC & Voltage Level Det. (SVLD)
- External voltage compare from PA[4] input possible (low resolution 4 bit AD converter)
-7 different levels from 2 V to 3.0 V for SVLD
- Used for Power Check after POR (2.0V check)
- Busy flag during measure
- Interrupt generated if SVLD measurement low

❑ 10-Bit Universal Counter
- 10, 8, 6 or 4 bit up/down counting
- Parallel load
- Event counting (PA[1] or PA[3/4])
- 8 different input clocks
- Full 10 bit or limited (8, 6, 4 bit) compare function
- 2 interrupt requests (on compare and on 0)
- Hi-frequency input on PA[1] and PA[3/4] or CPUClk
- Pulse width modulation (**PWM**) output

❑ Interrupt Controller
- 2 external and 6 internal interrupt request sources
- Each interrupt request can individually be masked
- Each interrupt flag can individually be reset
- Automatic reset of each interrupt request after read
- General interrupt request to CPU can be disabled
- Automatic enabling of general interrupt request flag when going into HALT mode

❑ Sleep Counter Reset (SCR)
- wake up the EM6580 from sleep mode
- 4 timings selectable by register
- Inhibit SCR by register

❑ Package form available
- SO-8/14
- Die form (9 pin possible due to additional I/O pin)

NB: All frequencies written in this document are related to a typical system clock of 32 kHz !

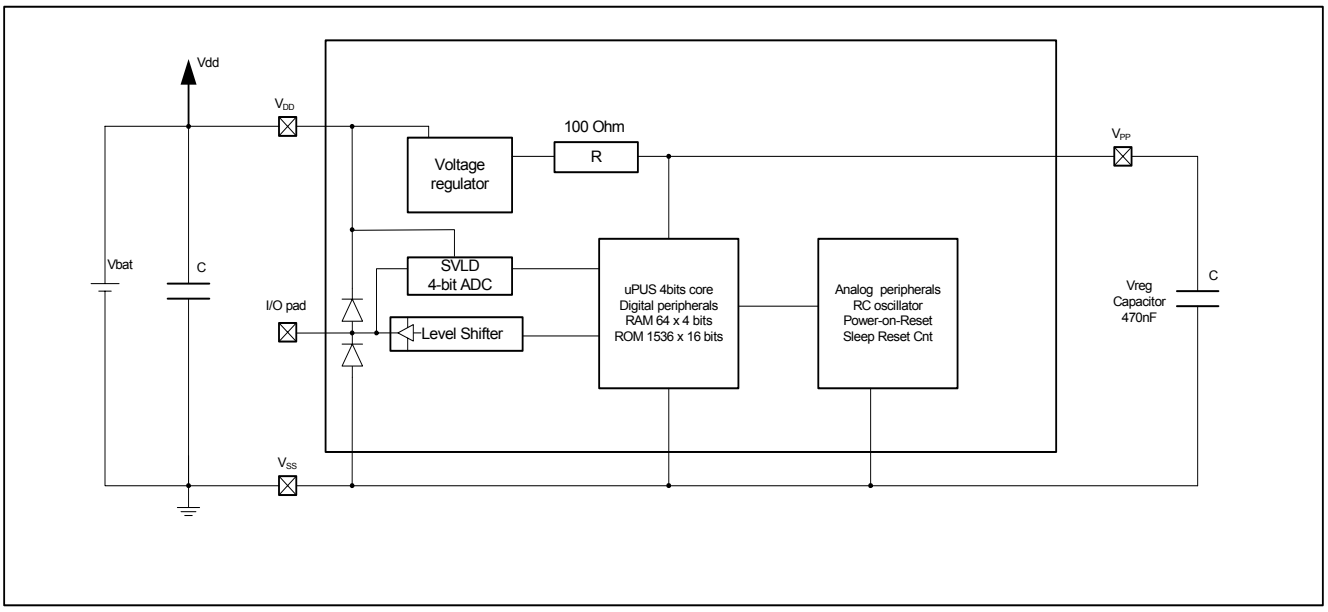
Table of Contents

1. Pin Description for EM6580

Table 1 EM6580 pin descriptions

# On Chip	SO-8	Signal Name	Description
1	1	PA0	general I/O, serial In, Wake-Up on Change, IRQ source,…
2	2	PA1	general I/O, serial CLK, timer source, external clock
3	3	PA2	general I/O, serial Out, freq., CPU reset status output,…
4	4	PA3	general I/O, serial Rdy/Cs, Interrupt source, reset
5	5	V_{ss}	ground – negative supply pin
6	6	PA4	general I, Reset, timer source, Interrupt source, Wake-Up, Compare I
7*	NC	PA5	general I/O, Wake-Up on Change, IRQ source
8	7	V_{REG}	regulated voltage supported by 470nF tw. V_{ss} High Voltage pin for Flash programming
9	8	V_{DD}	positive supply pin – capacitance tw. V_{DD} (C depends on V_{DD} noise)

Figure 3. Typical configuration for V_{dd} > 2.0V



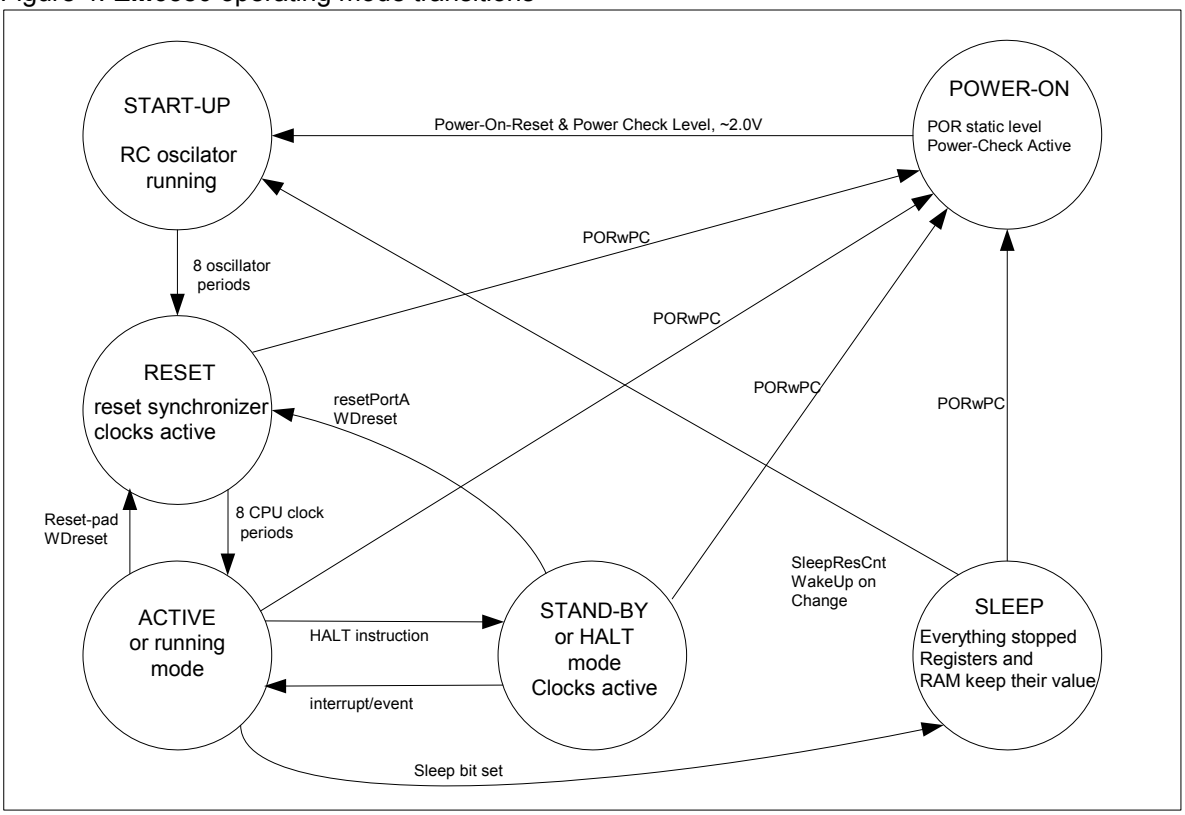
On I/O pins there are protective diodes towards V_{DD} and V_{SS}.

2. Operating modes

The EM6580 can operate in three different modes of which 2 are low-power dissipation modes (Stand-By and Sleep). The modes and transitions between them are shown in Figure 5.

1.) **Active** mode
2.) **Stand-By** mode
3.) **Sleep** mode

Figure 4. **EM6580** operating mode transitions



2.1 ACTIVE Mode

The active mode is the actual CPU running mode. Instructions are read from the internal ROM and executed by the CPU. Leaving the active mode: via the halt instruction to go into standby mode, writing the **SLEEP** bit to go into Sleep mode or detecting the reset to go into reset mode.

2.2 STANDBY (Halt) Mode

Executing a HALT instruction puts the EM6580 into standby mode. The voltage regulator, oscillator, watchdog timer, interrupts, timers and counters are operating. However, the CPU stops since the clock related to instruction execution stops. Registers, RAM and I/O pins retain their states prior to STANDBY mode. STANDBY is cancelled by a RESET or an Interrupt request if enabled.

2.3 SLEEP Mode

Writing to the **Sleep** bit in the **RegSysCntl1** register puts the EM6580 in sleep mode. The oscillator stops and most functions of the EM6580 are inactive. To be able to write to the **Sleep** bit, the **SleepEn** bit in **RegSysCntl2** must first be set to "1". In SLEEP mode only the voltage regulator is active to maintain the RAM data integrity, the peripheral functions are stopped and the CPU is reset.all other functions are in reset state. SLEEP mode may be cancelled by Wake/Up on change, external reset or by Sleep Reset Counter if any of them is enabled.

Waking up from sleep mode may takes some time to guarantee stable oscillation. Coming back from sleep mode puts the EM6681 in reset state and as such reinitializes all registers to their reset value. During sleep mode and the following start up the EM6580 is in reset state. Waking up from sleep mode clears the **Sleep** flag but not the **SleepEn** bit. Inspecting the **SleepEn** allows to determine if the EM6580 was powered up (**SleepEn** = "0") or woken from sleep mode (SleepEn = "1").

Table 2.3.1 Shows the Status of different EM6580 blocks in these three main operating modes.

Peripheral /// EM6580 mode	**ACTIVE** mode	**STAND-BY** mode	**SLEEP** mode
POR (static)	On	On	On
Voltage regulator	On	On	On (Low-Power)
RC-oscillator	On	On	Off
Clocks (Prescaler & RC divider)	On	On	Off
CPU	Running	In HALT – Stopped	Stopped
Peripheral register	"On"	"On" retain value	retain value
RAM	"On"	retain value	retain value
Timer/Counter	"On"	"On" if activated before	stopped
Supply Voltage Level Det.=SVLD	can be activated	can not be activated	Off
PortA / Reset pad debounced	Yes	Yes	No
Interrupts / events	Yes - possible	Yes - possible	No – not possible
Watch-Dog timer	On / Off (soft selectable)	On / Off (soft selectable)	No
Wake Up on Change PortA	No	No	On/Off (soft select.)
Sleep Reset Counter	Off	Off	On/Off (soft select.)

3. Power Supply

The EM6580 is supplied by a single external power supply between V_{DD} (V_{BAT}) and V_{SS} (ground). A built-in voltage regulator generates V_{REG} providing regulated voltage for the oscillator and the internal logic. The output drivers are supplied directly from the external supply V_{DD}. Internal power configuration is shown in Figure 3.

The internal voltage regulator is chosen for high voltage systems. It saves power by reducing the internal core logic's power supply to an optimum value. However, due to the inherent voltage drop over the regulator the minimal V_{DD} value is restricted to 2.0V .

4. Reset

Figure 6. illustrates the reset structure of the EM6580. One can see that there are five possible reset sources :

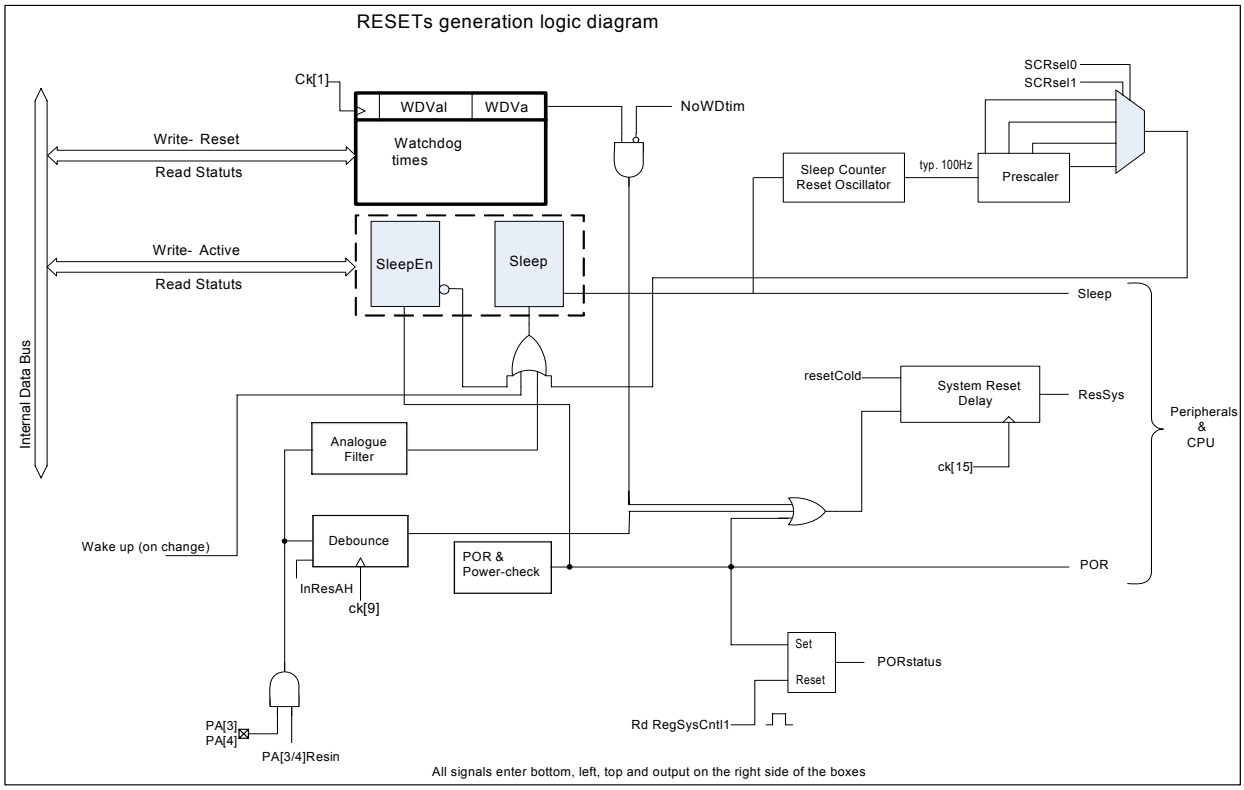
(1) Internal initial Power On Reset (POR) circuitry with Power-Check.→ POR, ResetCold, System Reset, Reset CPU
(2) External reset from PA[3/4] if software enabled → System Reset, Reset CPU
(3) Internal reset from the Digital Watchdog. → System Reset, Reset CPU
(4) Internal reset from the Sleep Counter Reset. → System Reset, Reset CPU
(5) Wake-Up on change from PA[0/5] or PA[3/4] if software enabled. → System Reset, Reset CPU

Table 4.1 Reset sources that can be used in different Operating modes

Reset Sources	**ACTIVE** mode	**STAND-BY** mode	**SLEEP** mode
POR (static) with Power Check	**Yes**	**Yes**	**Yes**
Software enabled reset on PA[3/4]	**XS** dig. debounced	**XS** dig. debounced	**XS** analog debounced
Digital Watch-Dog Timer	**XS**	**XS**	**No**
Sleep Counter Reset	**No**	**No**	**XS**
Wake Up on Change from Sleep	**No**	**No**	**XS**
Going in Sleep mode	**Yes**	**No**	**No**

XS = software enable

Figure 4. **EM6580** Reset Structure



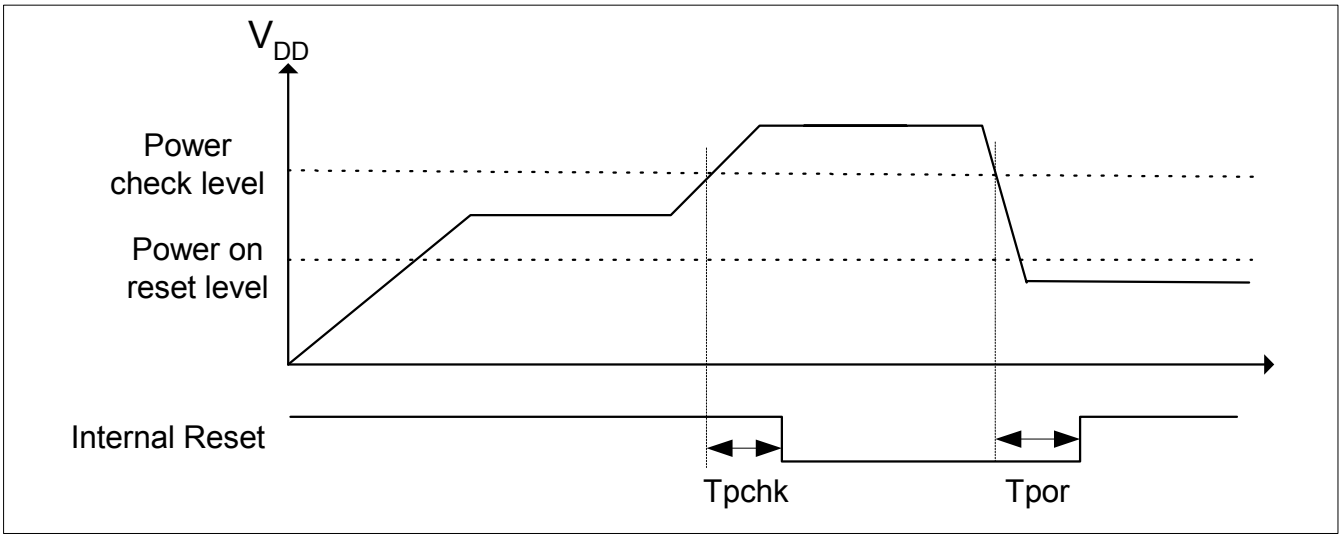
All reset sources activate the System Reset (ResSys). The 'System Reset Delay' ensures that the system reset remains active long enough for all system functions to be reset (active for 12 system clock cycles. CPU is reset by the same reset

As well as activating the system reset, the POR also resets all bits in registers marked '**p**' and the sleep enable (**SleepEn**) latch. System reset does not reset these register bits, nor the sleep enable latch.

4.1 POR with Power-Check Reset

An on chip detection circuit generates a power-on reset (POR). The POR is generated whenever VDD is below the detection level defined in section 17.4. A second detection circuit circuitry, power check level, ensures a proper start-up of the microcontroller if VDD is higher than this level. This level is higher than POR and defined in section 17.4.
If VDD drops below the static POR level, the chip goes in reset state.
To distinguish between POR reset and all other types of reset, the **PORstatus** bit in **RegSysCntl2** is set on at every POR and is cleared by writing the **RegSysCntl1** register.

Figure 5. EM6580 *Start-up sequence*



4.2 Input Port A Reset

By writing the **PA[3/4]ResIn** in **RegFreqRst** registers the PA[3] or PA[4] input becomes dedicated for external reset. This bit is cleared by POR only. Which input is selected is set by **IrqPA[3l/4h]** bit from **RegPACntl2** register which is described in Chapter 6.

Bit **InResAH** in the **RegFreqRst** register selects the PA[3/4] reset function in Active and standby (Halt) mode. If set to '0' the PA[3/4] reset is inhibited. If Set to '1' than PA[3/4] input goes through a debouncer and needs to respect timing associated with the debounce clock selection made by **DebSel** bit in **RegPresc** register.

This **InResetAH** bit has no action in sleep mode, where a Hi pulse on PA[3/4] always immediately triggers a system reset (only small analogue debouncer is attached to filter 1 or 2 μs spikes).

Overview of control bits and possible reset from PA[3] or PA[4] is specified in table 4.2.1 below.

Table 4.2.1 Possible Reset from PA[3] or PA[4]

PA[3/4]ResIn	**InResAH**	**ACTIVE** or **STAND-BY** mode	**SLEEP** mode
0	X	NO reset from PA[3] or PA[4]	NO reset from PA[3] or PA[4]
1	0	NO reset from PA[3] or PA[4]	Reset with small analog filter
1	1	Debounce reset with debck of * Ck[14]/ Ck[11]/ Ck[8] needing 0.25 ms / 2 ms / 16ms Hi pulse typ.	Reset with small analog filter

***** Ck[14]/ Ck[11]/ Ck[8] are explained in chapter 5.2 Prescaler.

4.3 Digital Watchdog Timer Reset

The Digital Watchdog is a simple, non-programmable, 2-bit timer, that counts on each rising edge of Ck[1]. It will generate a system reset if it is not periodically cleared. The watchdog timer function can be inhibited by activating an inhibit digital watchdog bit (**NoWDtim**) located in **RegVLDCntl**. At power up, and after any system reset, the watchdog timer is activated.

If for any reason the CPU stops or stays in a loop where watchdog timer is not periodically cleared, it activates the system reset signal. This function can be used to detect program overrun, endless loops, etc. For normal operation, the watchdog timer must be reset periodically by software at least every 2.5 seconds (system clock = 32 KHz), or a system reset signal is generated.

The watchdog timer is reset by writing a '1' to the **WDReset** bit in the timer. This resets the timer to zero and timer operation restarts immediately. When a '0' is written to **WDReset** there is no effect. The watchdog timer also operates in standby mode and thus, to avoid a system reset, standby should not be active for more than 2.5 seconds.

From a System Reset state, the watchdog timer will become active after 3.5 seconds. However, if the watchdog timer is influenced from other sources (i.e. prescaler reset), then it could become active after just 2.5 seconds. It is therefore recommended to use the Prescaler **IRQHz1** interrupt to periodically reset the watchdog every second.

It is possible to read the current status of the watchdog timer in **RegSysCntl2**. After watchdog reset, the counting sequence is (on each rising edge of CK[1]) : '00', '01', '10', '11', {WDVal1 WDVal0}). When reaching the '11' state, the watchdog reset will be active within ½ second. The watchdog reset activates the system reset which in turn resets the watchdog. If the watchdog is inhibited it's timer is reset and therefore always reads '0'.

Table 4.3.1 Watchdog timer register **RegSysCntl2**

Bit	**Name**	**Reset**	**R/W**	**Description**
3	WDReset	0	W	Reset the Watchdog (The Read value is always '0') 1 → Resets the Logic Watchdog 0 → no action
2	*SleepEn*	*0*	*R/W*	*See Operating modes (sleep)*
1	WDVal1	0	R	Watchdog timer data 1/4 ck[1]
0	WDVal0	0	R	Watchdog timer data 1/2 ck[1]
3	PORstatus	1 P*	R	Power-On-Reset status

1 P* POR sets the PORstatus bit which is cleared by writing register **RegSysCntl1**

4.4 Sleep Counter Reset

To profit the most from Low Power Sleep Mode and still supervise the circuit surrounding, one can enable the Sleep Counter Reset which only runs in Sleep mode and periodically wakes up the EM6580. Four (4) different Wake-Up periods are possible as seen in table below.

Control bits **SleepCntDis** which is set to default '0'by POR enables the Sleep Counter when the circuit goes into Sleep mode. The **SCRsel1**, **SCRsel0** bits that are used to determine Wake-Up period are in the **RegSleepCR** register. To disable the Sleep Counter in Sleep mode **SleepCntDis** must be set to '1'.

Table 4.4.2 Register **RegSleepCR**

Bit	Name	Reset	R/W	Description
3	NoPullPA[4]	0 por	R/W	Remove pull-up/down from PA[4] input
2	SleepCntDis	0 por	R/W	Disable Sleep Reset Counter when Hi
1	SCRsel1	0 por	R/W	Selection bit 1 for Sleep RCWake-Up period
0	SCRsel0	0 por	R/W	Selection bit 0 for Sleep RCWake-Up period

Table 4.4.3 Wake-Up period from Sleep selection

SCRsel1	SCRsel0	Sleep Reset Counter periods
0	0	1.5 internal low speed RC clock periods
0	1	15.5 internal low speed RC clock periods
1	0	127.5 internal low speed RC clock periods
1	1	1023.5 internal low speed RC clock periods

Refer to table 18.8 or the actual SCR timeout period timings

Sleep Counter Reset (SCR) uses the same prescaler (see chapter 5.3) as the System Clock in Active and StandBy mode. Prescaler reset is made automatically just before going into Sleep mode if SCR is enable. This causes the Sleep Reset Counter to have its specified period.

4.5 Wake-Up on Change

By writing the **WUchEn[0/5]** and/or **WUchEn[3/4]** bit in **RegPaCntl2** registers the PA[0] or PA[5] and/or PA[3] or PA[4] can generate a reset from sleep on any polarity change on a selected pin. The port selection is defined with bits **IrqPA[0l/5h]** and **IrqPA[3l/4h]**. See chapter 6 and Figure 10 for more details.

4.6 The CPU State after Reset

Reset initializes the CPU as shown in Table 4.6.1 below.

Table 4.6.1 Initial CPU value after Reset.

Name	Bits	Symbol	Initial Value
Program counter 0	12	PC0	$000 (as a result of Jump 0)
Program counter 1	12	PC1	Undefined
Program counter 2	12	PC2	Undefined
Stack pointer	2	SP	SP[0] selected
Index register	7	IX	Undefined
Carry flag	1	CY	Undefined
Zero flag	1	Z	Undefined
Halt	1	HALT	0
Instruction register	16	IR	Jump 0
Periphery registers	4	Reg.....	See peripheral memory map

5. Oscillator and Prescaler

5.1 RC Oscillator or external Clock

EM6580 can use the internal RC oscillator or external clock source for its operation.

The built-in RC oscillator without external components generates the system operating clock for the CPU and peripheral blocks. The RC oscillator is supplied by the regulated voltage.

The RC oscillator frequency can be chosen from 5 possibilities with a register with 2 basic frequencies.

These are typically 32kHz, 64kHz, 128kHz, 256kHz or 500kHz for 32kHz basic frequency or *50*kHz, *100*kHz, *200*kHz, *400*kHz or *800*kHz for *50*kHz basic frequency . Depending on the selected RC frequency. A special 4 stage freq. divider is available to be able to deliver to Prescaler which generates all System clock except CPU clock, a frequency close to 32 or *50* kHz to keep the peripheral timing as close as possible to specifications.

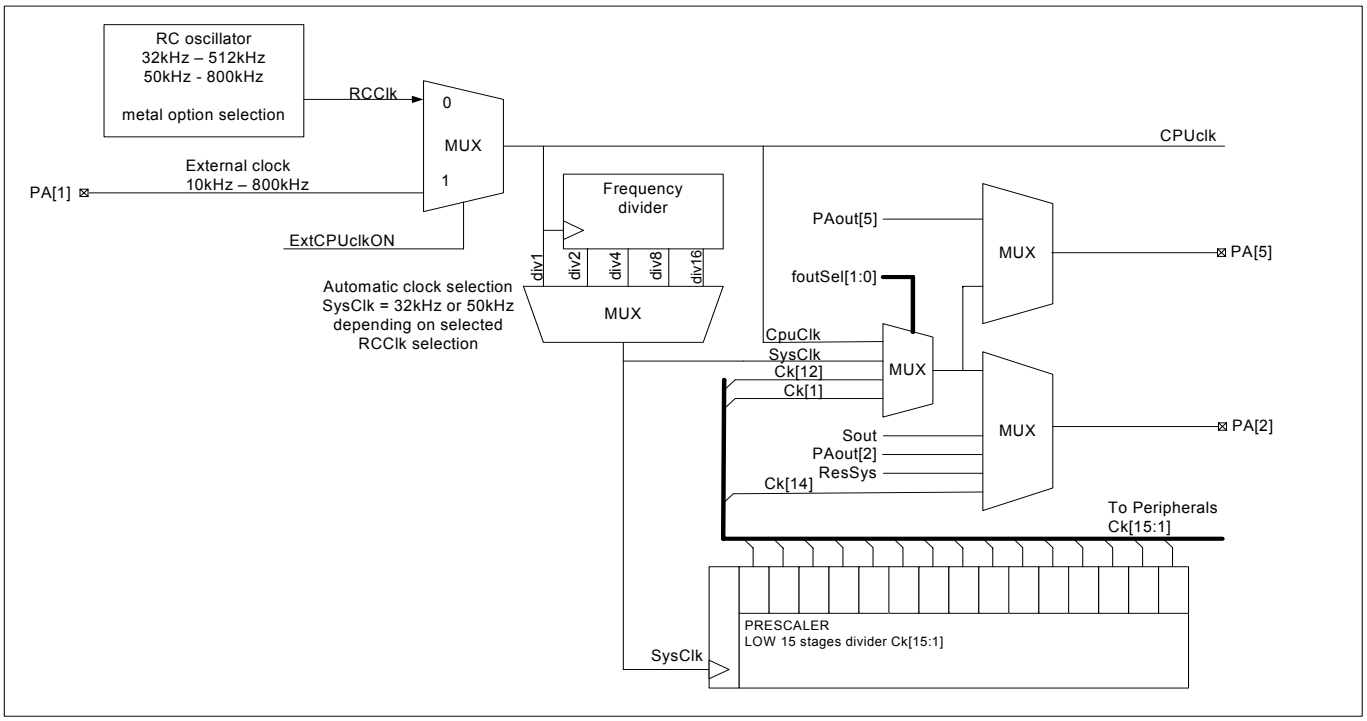
After POR the circuit always starts with the internal RC oscillator, but it can be switched to the external clock by setting the **ExtCPUclkON** bit in the register **RegPresc.** The external clock is input at PA[1] and must be in range from min. 10Khz to max. 1MHz. With this external frequency input all timing for peripherals change and the special 4 stage freq. divider **must** be adapted to best suit the applied external frequency to keep 32/*50*kHz System clock as close as possible. The system clock **must** be less than 64kHz. The external clock source must be a square wave with full amplitude from V_{ss} to V_{dd}. See Table 5.2.2 for advised special divisions depending on the external clock frequency.

Switching from internal RC oscillator to External clock or back from External clock to RC oscillator is made without generating a glitch on the internal clock. Once the circuit is running on the external Clock one can disable the RC oscillator by setting the **RCoscOff** bit in **RegSCntl2** to '1'.

In sleep mode the oscillator is stopped. It can be stopped also by setting the **RCoscOff** bit. This bit can be set only if **ExtCPUClkOn** was set before, indicating that the CPUck was switched from the internal RC oscillator to the external clock which MUST be present. If the External Clock stops without going into Sleep mode first the EM6580 can block and only POR can reset it.

Figure 8. below shows the connection of the RC oscillator and external clock and generation of CPUclk and System clock = SysClk which is divided by the special 4 stage Freq. Divider if needed as described in 5.2 and prescaler described in 5.3.

Figure 6. Clock source for CPU or system peripherals



5.2 Special 4 stage Frequency Divider

If an internal RC clock or external frequency higher than 32 kHz or *50* kHz is selected, then the special 4 stage Frequency Divider must be used to select a frequency close to 32 kHz or *50* kHz for the SysClk - system clock used by the Prescaler. This is done by register option.(Refer to RegMFP1 table chapter 14.8)

If the external clock will be used, the same register option used to select the division for SysClk. Table below shows recommended divisions in this 4 stage divider.

Table 5.2.1 PA[1] I/O status depending on its **RegPACntl3** and **RegPa0OE** registers

Ext. clock	**RC frequency**		**RC frequency or External freq. MUST be divided by**	**Typical Obtained SysClk Min.- typ. If RC - Max. [kHz]**
	(1)base	(2) base		
10 kHz – 50 kHz	32	*50*	No Division to SysClk	10 – 32 – 50
55 kHz – 100 kHz	64	*100*	Divided by **2**	27.5 – 32 – 50
110 kHz – 200 kHz	128	*200*	Divided by **4**	27.5 – 32 – 50
220 kHz – 400 kHz	256	*400*	Divided by **8**	27.5 – 32 – 50
400 kHz – 1 MHz	500	*800*	Divided by **16**	10 – 32 – 62.5

5.3 Prescaler

The prescaler consists of a fifteen element divider chain which delivers clock signals for the peripheral circuits such as timer/counter, debouncer and edge detectors, as well as generating prescaler interrupts. The input to the prescaler is the system clock signal closest to 32 kHz or *50* kHz which comes from the RC oscillator or external clock as divided by the preceding divider. Power on initializes the prescaler to Hex(0001).

Table 5.3.1 Prescaler Clock Name Definition

Function	**Name**	**32 KHz SysClk**	***50* KHz SysClk**
System clock	Ck[16]	32768 Hz	*50000* Hz
System clock / 2	Ck[15]	16384 Hz	*25000* Hz
System clock / 4	Ck[14]	8192 Hz	*12500* Hz
System clock / 8	Ck[13]	4096 Hz	*6250* Hz
System clock/ 16	Ck[12]	2048 Hz	*3125* Hz
System clock / 32	Ck[11]	1024 Hz	*1562* Hz
System clock / 64	Ck[10]	512 Hz	*781* Hz
System clock / 128	ck [9]	256 Hz	*390* Hz

Function	**Name**	**32 KHz SysClk**	***50* KHz SysClk**
System clock / 256	Ck[8]	128 Hz	*195* Hz
System clock / 512	Ck[7]	64 Hz	*97* Hz
System clock / 1024	Ck[6]	32 Hz	*49* Hz
System clock / 2048	Ck[5]	16 Hz	*24* Hz
System clock / 4096	Ck[4]	8 Hz	*12* Hz
System clock / 8192	Ck[3]	4 Hz	*6* Hz
System clock / 16384	Ck[2]	2 Hz	*3* Hz
System clock / 32768	Ck[1]	1 Hz	*1.5* Hz

Figure 7. Prescaler Frequency Timing

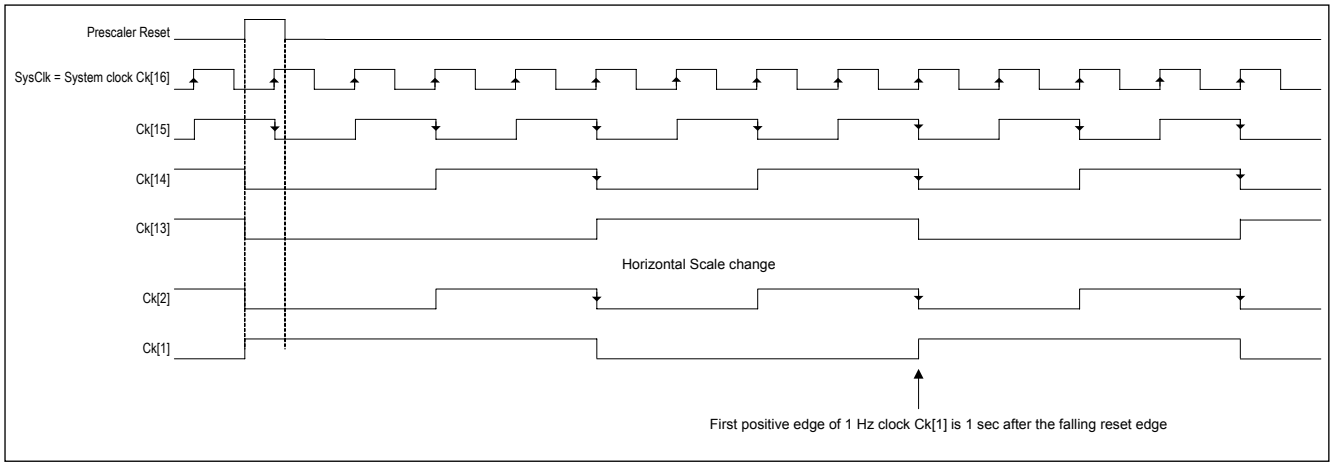


Table 5.3.2 Control of Prescaler Register **RegPresc**

Bit	Name	Reset	R/W	Description
3	ExtCPUclkON	*p*	R/W	Ext. Clock selection instead of RC oscillator for CPUClk.
2	ResPresc	0	R/W	Write Reset prescaler **1** → Reset the divider chain from Ck[14] to Ck[2], sets Ck[1]. **0** → No action. The Read value is always '0'
1	PrIntSel	0	R/W	Prescaler Interrupt select. **0** → Interrupt from Ck[4] (typ. 8/*12* Hz) **1** → Interrupt from Ck[7] (typ. 64/*97* Hz)
0	DebSel	0	R/W	Debouncer clock select. **0** → Debouncer with Ck[8] **1** → Debouncer with Ck[11] or Ck[14]

*p** reset to '0' by POR only.

With DebSel = 1 one may choose either the Ck[11] or Ck[14] debouncer frequency by selecting the corresponding register mask option.(Refer to RegMFP1 table chapter 14.8) Relative to 32kHz the corresponding max. debouncer times are then 2 ms or 0.25 ms. For the register option selection refer to chapter 14.8

Switching the **PrIntSel** may generate an interrupt request. Avoid it with **MaskIRQ64/8** = 0 selection during the switching operation.

The prescaler contains 2 interrupt sources:
- IRQ64/8 ; this is Ck[7] or Ck[4] positive edge interrupt, the selection is depending on bit **PrIntSel.**
- IRQHz1 ; this is Ck[1] positive edge interrupt

There is no interrupt generation on reset.

The first IRQHz1 Interrupt occurs typically. 1 sec (if SysClk = 32kHz) after reset. (*0.65 sec if SysClk is 50kHz*).

NOTE: If not written explicitly all timing in peripherals is calculated for 32 kHz System Clock !

6. Input and Output port A

The EM6580 has:
- port PA[3:0], one 4-bit input/output port
- port PA[4], one 1 bit input port.
- port PA[5], one 1 bit input/output. (not available in 8-pin package).

Pull-up and Pull-down resistors can be added to all these ports with register options.

6.1 Input / Output Port Overview

Table 6.1.1 Input and Output port overview

	PA[0]	**PA[1]**	**PA[2]**	**PA[3]**	**PA[4]**	***PA[5]****
Pin in 8pin package	1	2	3	4	6	NC*
General I/O	**I/O**	**I/O**	**I/O**	**I/O**	**I**	**I/O**
Serial interface	Sin **I**	Sclk **I/O**	Sout **O**	Rdy/CS **O**	--	--
WakeUp on change	yes* **I**	--	--	yes* **I**	yes* **I**	yes* **I**
Softw. pullUp/Down	Yes	yes	yes	yes	yes	yes
Register option pullUp/Down	--	--	--	--	yes	--
Timer input	--	yes **I**	--	yes* **I**	yes* **I**	--
Irq debounce & edge select.	yes* **I**	--	--	yes* **I**	yes* **I**	yes* **I**
CPU soft. variable input	yes* **I**	--	--	yes* **I**	yes* **I**	yes* **I**
Analogue compare Input	--	--	--	--	yes **I**	--
External reset input	--	--	--	yes* **I**	yes* **I**	--
External CPU clock input	--	yes **I**	--	--	--	--
PWM timer out	yes **O**	yes **O**	--	--	--	--
freq. Output (RC, 2kHz, 1Hz)	--	--	yes* **O**	--	--	yes* **O**
CPU reset condition	--	--	yes **O**	--	--	--

NC* – Pad PA[5] is Not Connected in 8-pin package, available on 14-pin and in die form
Register option – Refer to RegMFP1 table chapter 14.8

As shown in Figure 10, Logic for the Wake up on change reset which is possible only from Sleep mode, Debouce and IRQ function on Rising or falling edge are implemented only twice but can be attached and configured by registers to 4 different pads when used as inputs.

Ports PA[0] and PA[5] can be configured to have wakeup on Change, and debounced or non-debounced IRQ on the falling or rising edge. The same function is available on ports PA[3] or PA[4] which in addition can be dedicated to input reset . Registers **RegPACntl1** and **RegPACntl2** make this selection.

Table 6.1.2 Register **RegPACntl1**

Bit	Name	POR	R/W	Description
3	DebounceNoPA[3/4]	0	R/W	Debounce on when Low for PA[3/4] input
2	DebounceNoPA[0/5]	0	R/W	Debounce on when Low for PA[0/5] input
1	EdgeFallingPA[3/4]	0	R/W	IRQ edge selector for interrupt from PA[3/4] input
0	EdgeFallingPA[0/5]	0	R/W	IRQ edge selector for interrupt from PA[0/5] input

* Default is debouncer On and Rising edge for IRQ

Table 6.1.3 Register **RegPACntl2**

Bit	Name	POR	R/W	Description
3	WUchEnPA[3/4]	0	R/W	Wake/Up on change EN on PA[3] or PA[4]
2	WUchEnPA[0/5]	0	R/W	Wake/Up on change EN on PA[0] or PA[5]
1	IrqPA[3l/4h]	0	R/W	PA[3] if Low / PA[4] if High for IRQ source
0	IrqPA[0l/5h]	0	R/W	PA[0] if Low / PA[5] if High for IRQ source

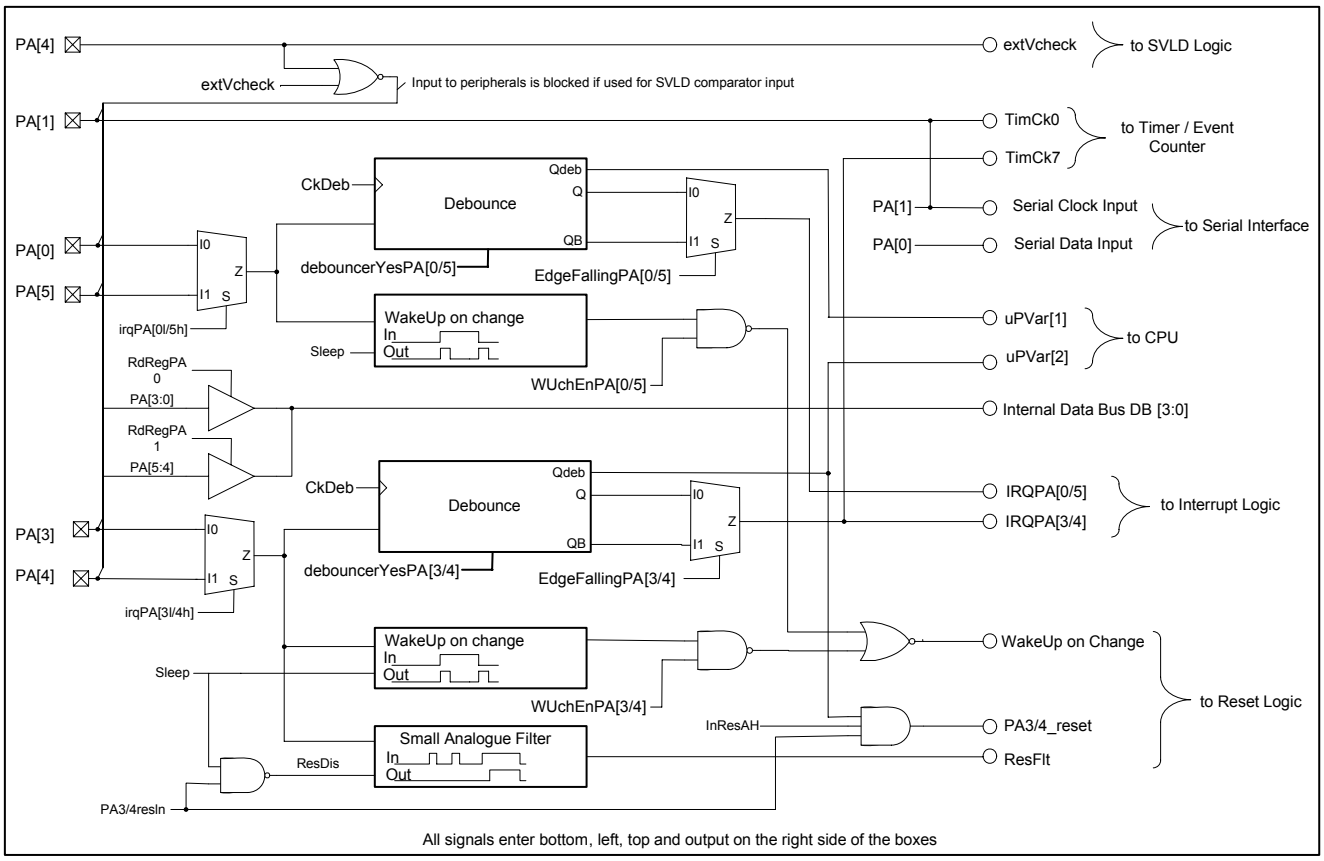
* Default: No wake Up on change and IRQ source, or reset and timer input, would be PA[3], PA[0]

6.2 PortA as Input and its Multiplexing

The EM6580 can have up to 6 1-bit general purpose CMOS input ports. The port A input can be read at any time, pull-up or pull-down resistors can be chosen by software.

Figure 10 explains how the inputs are treated with control signals and how they are distributed to different peripherals and the CPU. This is also listed in Table 6.1.1 Input and Output ports overview.

Figure 8. EM6580 Multiplexed Inputs diagram

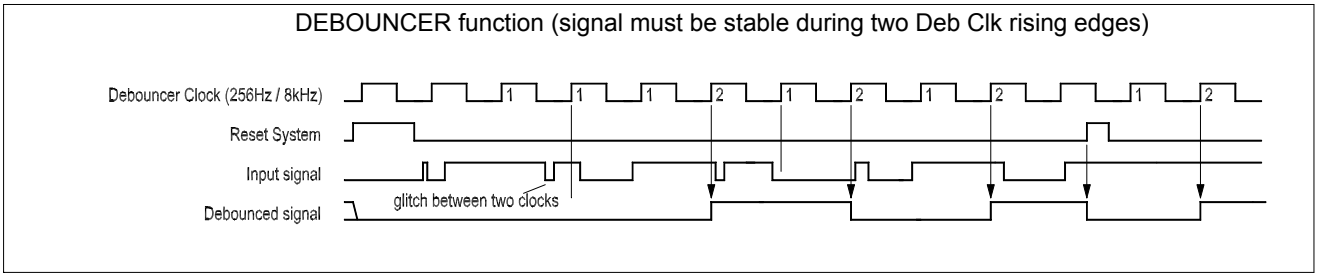


Some Input functions are explained below.

6.2.1 Debouncer

The debouncer is clocked with one of the possible debouncer clocks (Ck[14] / Ck[11] / Ck[8] and can be used only in Active or StandBy mode (as only in these two modes clocks are running). The input signal has to be stable on two successive debouncer rising clock edges and must not change between them.

Figure 9. Debouncer function



6.2.2 IRQ on Port A

For interrupt request generation (IRQ) one can choose direct or debounced input and rising or falling edge IRQ triggering. With the debouncer selected **debounceYesPA[x/y]**, the input must be stable for two rising edges of the selected debouncer clock CkDeb. This means a worst case of 16ms(default) or 2ms (0.25ms by metal mask) with a system clock of 32kHz.

Either a rising or falling edge on the port A inputs - with or without debouncing - can generate an interrupt request. This selection is done by **edgeFallingPA[x/y].**
PortA can generate max 2 different interrupt requests. Each has its own interrupt mask bit in the **RegIRQMask1** register. When an IRQ occurs, inspection of the **RegIRQ1** and **RegIRQ2** registers allow the interrupt to be identified and treated.
At power on or after any reset the **RegIRQMask1** is set to 0, thus disabling any input interrupt. A new interrupt is only stored with the next active edge after the corresponding interrupt mask is cleared. See also the interrupt chapter 9.
It is recommended to mask the port A IRQ's while one changes the selected IRQ edge. Otherwise one may generate an unwanted IRQ (Software IRQ). I.e. if a bit PA[0/5] is '0' then changing from positive to negative edge selection on PA[0/5] will immediately trigger an **IRQPA[0/5]** if the IRQ was not masked.

6.2.3 Pull-up/down

On Each terminal of PA[3:0] and PA[5] an internal pull-up (metal mask MAPU[n]) or pull-down (metal mask MAPD[n]) resistor can be connected per metal mask option. By default the two resistors are in place. In this case one can choose by software to have either a pull-up, a pull-down or no resistor.
See below for better understanding. If the port is used also as output please check Chapter 6.3.1 CMOS / Nch. Open Drain Output.
PA[4] can have only strong Pull-up or Pull-down resistor which can be removed by the software register **NoPullUpDown[4].**

PA[4] can have only strong Pull-up or Pull-down resistor. This resistor can be disconnected by software in register RegSleepCR bit **NoPullPA[4].**

For Metal mask selection and available resistor values refer to chapter 13.

Pull-down ON: MAPD[n] must be in place , with n=0, 1, 2, 3, 5
AND bit **NoPdPA[n]** must be '0' .

Pull-down OFF: MAPD[n] is not in place,
OR if MAPD[n] is in place **NoPdPA[n]** = '1' cuts off the pull-down.
OR selecting **NchOpDrPA[n]** = '1' cuts off the pull-down.

Pull-up ON : MAPU[n] must be in place,
AND bit **NchOpDrPA[n]** must be '1' ,
AND (bit **OEnPA[n]** = '0' (input mode) **OR** if **OEnPA[n]** = '1' while **PAData[n]** = 1.)

Pull-up OFF : MAPU[n] is not in place,
OR if MAPU[n] is in place **NchOpDrPA[n]** = '0' cuts off the pull-up,
OR if MAPU[n] is in place and if **NchOpDrPA[n]** = '1' then **PAData[n]** = 0 cuts off the pull-up.

Never pull-up and pull-down can be active at the same time.

Any port A input must never be left open (high impedance state, not connected, etc.) unless the internal pull resistor is in place (mask option) and switched on (register selection). Any open input may draw a significant cross current which adds to the total chip consumption.

Note: *The mask settings_MAPU[n]) and MAPD[n] do not define the default pull direction, but the pull possibilities. It is the software which defines the pull direction (pull-up or pull-down). The only exception is on PA[4] where the user must decide if he wants pull-up or pull-down. For this input the selected pull direction will always be valid unless the software disconnects the pull resistor.*

6.2.4 Software test variables

As shown in Figure 10 PA[0/5] or PA[3/4] are also used as input conditions for conditional software branches. These CPU inputs are always debounced and non-inverted.

- debounced PA[0/5] is connected to CPU TestVar1
- debounced PA[3/4] is connected to CPU TestVar2

CPU TestVar3 is connected to Ground and can not be used in Software.

6.2.5 Port A for 10-Bit Counter

The PA[1] and PA[3/4] inputs can be used as the clock input terminal for the 10 bit counter

- PA[1] is at counter clock selection 0. The input is direct (no debouncer is possible
- PA[3/4] is at counter clock selection 7. As for the IRQ generation, debounced or input directly and non-inverted or inverted input is possible. This is defined with the register **RegPaCntl1**. Debouncing the input is always recommended.

6.2.6 Port A Wake-Up on change

In sleep mode if configured port PA[0/5] or PA[3/4] inputs are continuously monitored to wake up on change, which will immediately wake up the EM6580.

6.2.7 Port A for Serial Interface

When the serial interface is used in slave mode, PA[0] is used for serial data input and PA[1] for the serial clock.

6.2.8 Port A for External Reset

In Active and Stand-by (Halt) mode a positive debounced pulse on PA[3/4] can be the source of a reset when **PA[3/4]ResIn** and **InResAH** are set at '1'. When **IrqPA[3l/4h]** is '0' than PA[3] is selected for Reset source and when **IrqPA[3l/4h]** is '1' than PA[4] is selected for Reset source.

6.2.9 Port PA[4] as Comparator Input

When using the PA[4] as an input to the internal SVLD comparator NO pull resistor should be connected on this terminal. Otherwise the device may draw excessive current.

First PA[4] pull-up/down resistor should be disconnected by software and the **ExtVcheck** bit can be set to '1'. This dedicates PA[4] as SVLD resistor divider input to the SVLD comparator.

At this point the measurements respect the same timing as any other SVLD measurements as explained in Chapter Supply Voltage Detector. It can also generate an IRQ if the input voltage is lower as Comparator level. Thus configured a direct read of PA[4] will result in reading '0'.

6.2.10 Reset and Sleep on Port A

During circuit initialisation, all Control registers are reset by Power On Reset and therefore all pull-ups are off and all pull-downs are on.
During Sleep mode, the circuit retains its register values. As such the PA configurations remain active also during Sleep.Sleep mode is cancelled with any Reset. However the Reset State does not reset the Control registers bits which are specifically marked to be initialised by POR only. (Pull, Nch Open drain, Freq out, etc configurations).
however Sleep mode is cancelled with a Reset and all system register will be reinitialized at this point. (the circuit is in Reset State. and pull-downs, if previously turned on. After any reset the serial interface parameters are reset to : Slave mode, Start and Status = 0, LSB first, negative edge shift , PA[3:0] tri-state.

6.2.11 Port A Blocked Inputs

In sleep mode if PortA inputs are not used and prepared for Wake-Up on Change or Reset these inputs are blocked. At that time port can be undefined from external and this will not generate an over-consumption.

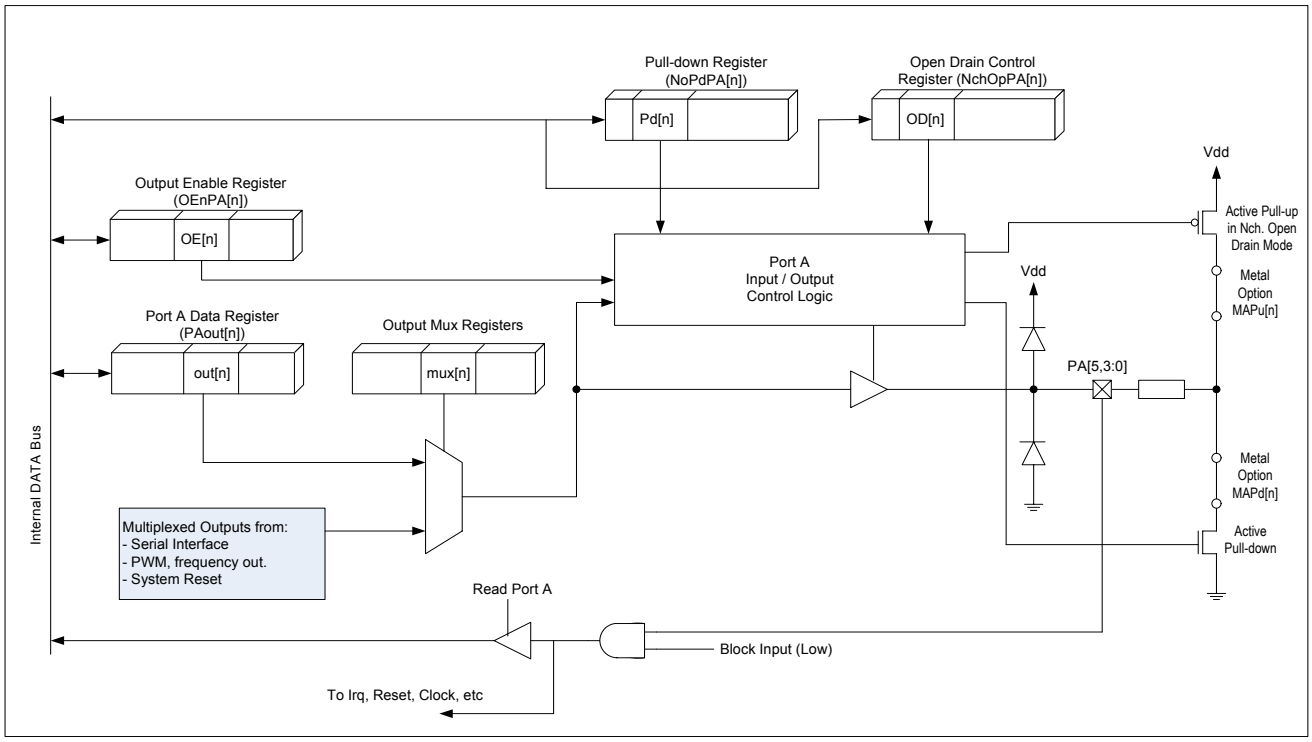
PA[0] : Blocked if Sleep bit set and no IRQ or Wake-up defined on this input
PA[1] : Blocked if Sleep bit set
PA[2] : Blocked if Sleep bit set
PA[3] : Blocked if Sleep bit set and no IRQ or Wake-up defined on this input
PA[4] : Blocked if Sleep bit set and no IRQ or Wake-up defined on this input
Also blocked if External VLD check enabled
PA[5] : Never blocked

6.3 PortA as Output and its Multiplexing

The EM6580 can have up to 4 (5 in Die form or 14-pin package) bit general purpose CMOS or N-channel Open Drain Output ports. Table 6.1.1 Input and Output ports overview shows all the possibilities. Figure 12 shows the output architecture and possible output signals together with software controlled pull-up and pull-down resistors which are disconnected when the port is an output and in a defined state, to preclude additional consumption.

The output multiplexing registers are **RegPACntl3** and **RegPACntl4**.

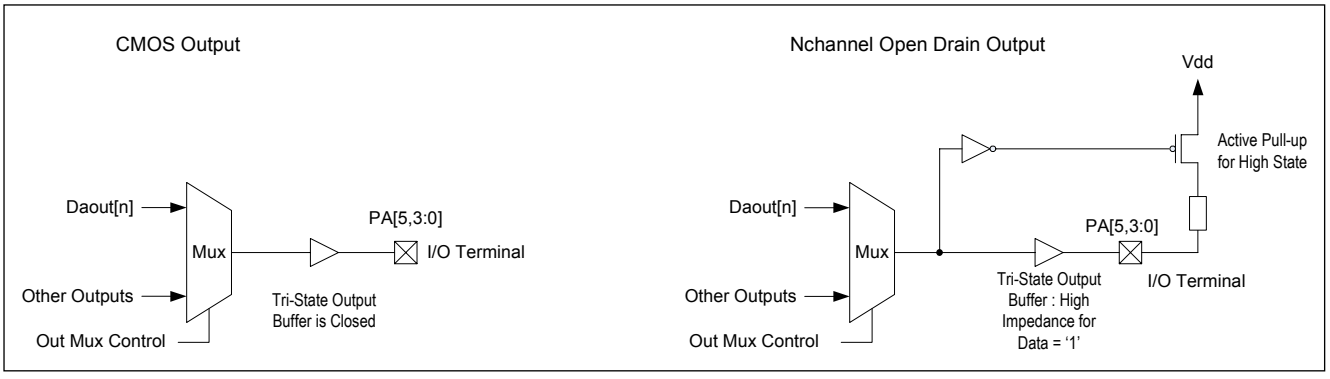
Figure 10. Port A Architecture (Outputs)



6.3.1 CMOS / Nch. Open Drain Output

The port A outputs can be configured as either CMOS or Nch. open drain outputs. In CMOS both logic '1' and '0' are driven out on the terminal. In Nch. Open Drain only the logic '0' is driven on the terminal, the logic '1' value is defined by the internal pull-up resistor (if implemented), or high impedance.

Figure 11. CMOS or Nch. Open Drain Outputs



NOTE: State of I/O pads may not be defined until V_{reg} reaches typ. 0.8V and Power-On-Reset logic supplied by V_{reg} clears them to Inputs.
This time depends on how fast capacitor on V_{reg} is charged and typ. it can be in range of couple of ms.

6.4 Port A registers

The two Control registers for Input control, **RegPACntl1** and **RegPACntl2,** were already shown in chapter 6; Input / Output Ports Overview.

Table 6.4.1 Register **RegPA0**

Bit	Name	Reset	R/W	Description
3	PAData[3]	0	R* /W	PA[3] input and PAout[3] output
2	PAData[2]	0	R* /W	PA[2] input and PAout[2] output
1	PAData[1]	0	R* /W	PA[1] input and PAout[1] output
0	PAData[0]	0	R* /W	PA[0] input and PAout[0] output

* Direct read on Port A terminals

Table 6.4.2 Register **RegPa0OE**

Bit	Name	Reset	R/W	Description
3	OEnPA[3]	0	R/W	I/O control for PA[3] , output when OEnPA[3] = Hi
2	OEnPA[2]	***0*** P**	R/W	I/O control for PA[2] , output when OEnPA[2] = Hi
1	OEnPA[1]	0	R/W	I/O control for PA[1] , output when OEnPA[1] = Hi
0	OEnPA[0]	0	R/W	I/O control for PA[0] , output when OEnPA[0] = Hi

P** On Reset PA[2] is forced to output if (PA[0]='0', PA[1]='1', PA[4]='1', Sout/RstPA[2]='1' and freqOutPA[2]='1') until System reset is finished. Refer also to Table 6.4.11.
After Reset is finished and circuit starts to execute instructions PA[2] becomes tri-state input.
Bit OEnPA[2] is reset to '0' with every Reset.

Table 6.4.3 Register **Pa0noPDown**

Bit	Name	POR*	R/W	Description
3	NoPdPA[3]	0	R/W	No pull-down on PA[3]
2	NoPdPA[2]	0	R/W	No pull-down on PA[2]
1	NoPdPA[1]	0	R/W	No pull-down on PA[1]
0	NoPdPA[0]	0	R/W	No pull-down on PA[0]

POR* Reset only with Power On Reset

Table 6.4.4 Register **Pa0NchOpenDr**

Bit	Name	POR*	R/W	Description
3	NchOpDrPA[3]	0	R/W	Nch. Open Drain on PA[3]
2	NchOpDrPA[2]	0	R/W	Nch. Open Drain on PA[2]
1	NchOpDrPA[1]	0	R/W	Nch. Open Drain on PA[1]
0	NchOpDrPA[0]	0	R/W	Nch. Open Drain on PA[0]

* Reset only with Power On Reset, Default "0" is: CMOS on PA[3..0]

Table 6.4.5 Register **RegPA1**

Bit	Name	Reset	R/W	Description
3	NchOpDrPA[5]	p**	R* /W	Nch. Open Drain on PA[5]
2	OEnPA[5]	0	R* /W	I/O control for PA[5] , output when OEnPA[5] = Hi
1	PAData[5]	0	R* /W	PA[5] input and PAout[5] output
0	PAData[4]*	0	R*	PA[4] input

* Direct read on Port A terminals p** reset to '0' by POR only

Table 6.4.6 Register **RegFreqRst**

Bit	Name	POR	R/W	Description
3	InResAH	p	R/W	Input reset On in Active and StandBy mode
2	PA3/4resIn	p	R/W	PA3/4 dedicated for Input reset when set at '1'
1	foutSel[1]	x	R/W	Output Frequency selection (foutSel[1:0])
0	foutSel[0]	x	R/W	(11)CPUClk, (10) SysClk, (01) 2kHz, (00) 1Hz

Interrupt PortA Control bits **MaskIRQPA[0/5]** and **MaskIRQPA[3/4]** used to enable (Mask) the Interrupt ReQuest IRQ from PortA are in register **RegIRQMask1**.

Interrupt status bits **IRQPA[0/5]** and **IRQPA[3/4]** used to signal the Interrupt from PortA are in register **RegIRQ1**. They are both shown in Chapter Interrupt Controller.

Note: CPUClk = RCClk if no external clock used.

In case of external clock, CPUClk is equal to the PA[1] input clock.

Output multiplexing registers are shown below.

Table 6.4.7 Register **RegPACntl3**

Bit	Name	POR	R/W	Description
3	SerialStPA[3]	0	R/W	Output selection for PA[3] when output
2	SerialCkPA[1]	0	R/W	Output selection for PA[1] when output
1	PWMoutPA[1]	0	R/W	Output selection for PA[1] when output
0	PWMoutPA[0]	0	R/W	Output selection for PA[0] when output

Table 6.4.8 Register **RegPACntl4**

Bit	Name	POR	R/W	Description
3	NoPdPA[5]	0	R/W	No pull-down on PA[5]
2	freqOutPA[5]	0	R/W	Output selection for PA[5] when output
1	Sout/rstPA[2]	1	R/W	Output selection for PA[2] when output
0	freqOutPA[2]	1	R/W	Output selection for PA[2] when output

Table 6.4.9 PA[0] I/O status depending on its **RegPACntl3** and **RegPa0OE** registers

OEnPA[0]	PWMoutPA[0]		Description of PA[0] terminal
0	**X**		Input
1	**0**		PAout[0] general Output
1	**1**		PWM Output from the 10-Bit Counter

Table 6.4.10 PA[1] I/O status depending on its **RegPACntl3** and **RegPa0OE** registers

OEnPA[1]	SerialCkPA[1]	PWMoutPA[1]	Description of PA[1] terminal
0	**X**	**X**	Input
1	**0**	**0**	PAout[1] general Output
1	**0**	**1**	PWM Output from the 10-Bit Counter
1	**1**	**X**	Sclk (Serial interface clock output)

Table 6.4.11 PA[2] I/O status depending on its **RegPACntl4** and **RegPa0OE** registers

OEnPA[2]	Sout/rstPA[2]	freqOutPA[2]	Description of PA[2] terminal
0	X	X	Input
1	0	0	PAout[2] general Output
1	0	1	Freq. Output (CPUClk,SysClk, 2kHz, 1Hz)
1	1	0	Sout (Serial interface data output)
1	1	1	High level '1' during Reset state output 8kHz frequency output while out of reset state Low level '0' output during sleep
0 PA[0] = '1' PA[4] = '1' PA[1] = '0'	1	1	Output: high level during Reset state Input: out of reset state and during sleep

Frequency output is selected in 6.4.6 Register **RegFreqRst**

Table 6.4.12 PA[3] I/O status depending on its **RegPACntl3** and **RegPa0OE** registers

OEnPA[3]	SerialStPA[3]		Description of PA[3] terminal
0	**X**		Input
1	**0**		PAout[3] general Output
1	**1**		Rdy/CS (Serial interface status output)

Table 6.4.13 PA[5] I/O status depending on its **RegPACntl4** and **RegPA1** registers

OEnPA[5]	freqOutPA[5]		Description of PA[5] terminal
0	**X**		Input
1	**0**		PAout[5] general Output
1	**1**		Freq. Output (CPUClk,SysClk, 2kHz, 1Hz)

Frequency output is selected in 6.4.6 Register **RegFreqRst**

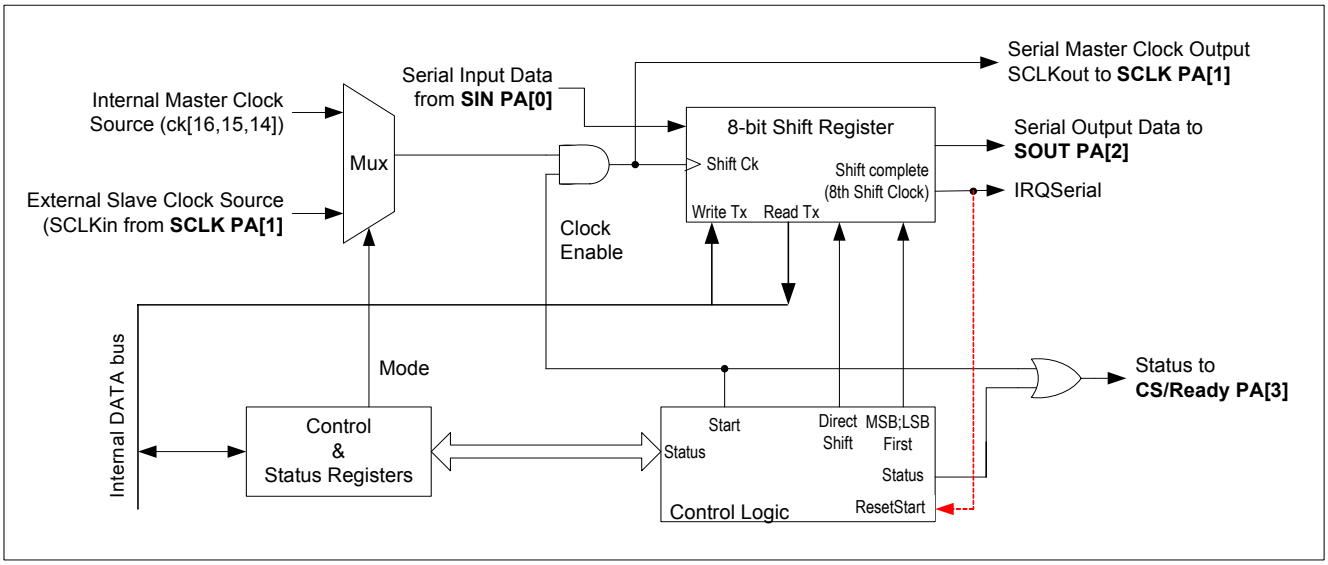
Note: CPUClk = RCClk if no external clock used.
In case of external clock, CPUClk is equal to the PA[1] input clock.

7. Serial Port

The EM6580 contains a simple, half duplex three wire synchronous type serial interface., which can be used to program or read an external EEPROM, ADC, ... etc. Its I/O are multiplexed on Port A.

For data reception, a shift-register converts the serial input data on the SIN(PA[0]) terminal to a parallel format, which is subsequently read by the CPU in registers **RegSDataL** and **RegSDataH** for low and high nibble. To transmit data, the CPU loads data into the shift register, which then serializes it on the SOUT(PA[2]) terminal. It is possible for the shift register to simultaneously shift data out on the SOUT terminal and shift data in on the SIN terminal. In Master mode, the shifting clock is supplied internally by the Prescaler : one of three prescaler frequencies are available, Ck[16], Ck[15] or Ck[14]. In Slave mode, the shifting clock is supplied externally on the SCLKIn(PA[1]) terminal. In either mode, it is possible to program : the shifting edge, shift MSB first or LSB first and direct shift output. All these selection are done in register **RegSCntl1** and **RegSCntl2**.

Figure 12. Serial Interface Architecture



The PA[3..0] terminal configuration is shown in Figure 10 and 12. When the Serial Interface is used then care should be taken not to use inputs and outputs needed for Serial Interface for other peripherals !:

- PA[0] {SIN} must be dedicated to Serial input if needed and can not be used for IRQ, Software Variable jumps or Output. It can be still used for Wake-Up on Change
- PA[1] {SCLK} is an output for Master mode {SCLKOut} and an input for Slave mode {SCLKIn}. But different functions can be Switched On/Off with care as they are needed.
- PA[2] {SOUT} must be dedicated to Serial Data Output if needed and can not be used for Analogue input, or other Output.
- PA[3] {CS/ Ready } if used for serial Interface status output. When used for Serial Interface it should not be used for IRQ, Software Variable jumps or Output. It can be still used for Wake-Up on Change.

Note:
Before using the serial interface, the corresponding circuit terminals must be configured accordingly.

7.1 General Functional Description

After power on or after any reset the serial interface is in serial slave mode with **Start** and **Status** set to 0, LSB first, negative shift edge and all outputs are in high impedance state.
When the **Start** bit is set, the shift operation is enabled and the serial interface is ready to transmit or receive data, eight shift operations are performed: 8 serial data values are read from the data input terminal into the shift register and the previous loaded 8-bits are send out via the data output terminal. After the eight shift operation, an interrupt is generated, and the **Start** bit is reset.

Parallel to serial conversion procedure (master mode example).

Setup the circuit IO's accordingly.
Write to **RegSCntl1** serial control (clock freq. in master mode, edge and MSB/LSB select).
Write to **RegSDataL** and **RegSDataH** (shift out data values).
Write to **RegSCntl2** (Start=1, mode select, status). → Starts the shift out
After the eighth clock an interrupt is generated, **Start** becomes low. Then, interrupt handling

Serial to parallel conversion procedure (slave mode example).

Setup the circuit IO's accordingly.
Write to **RegSCntl1** (slave mode, edge and MSB/LSB select).
Write to **RegSCntl2** (Start=1, mode select, status).
After eight serial clocks an interrupt is generated, **Start** becomes low.
Interrupt handling.
Shift register **RegSDataL** and **RegSDataH** read.
A new shift operation can be authorized.

7.2 Detailed Functional Description

Master or Slave mode is selected in the control register **RegSCntl1.**
In Slave mode, the serial clock comes from an external device and is input via the PA[1] terminal as a synchronous clock (SCLKIn) to the serial interface. The serial clock is ignored as long as the **Start** bit is not set. After setting **Start,** only the eight following active edges of the serial clock input PA[1] are used to shift the serial data in and out. After eight serial clock edges the **Start** bit is reset. The PA[3] terminal is a copy of the (**Start OR Status)** bit values, it can be used to indicate to the external master, that the interface is ready to operate or it can be used as a chip select signal in case of an external slave.

In Master mode, the synchronous serial clock is generated internally from the system clock. The frequency is selected from one out of three sources (**MS0** and **MS1** bits in **RegSCntl1**) **.** The serial shifting clock is only generated during **Start** = high and is output to the SCLK terminal as the Master Clock (SCLKOut). When **Start** is low, the serial clock output on PA[1] is 0.

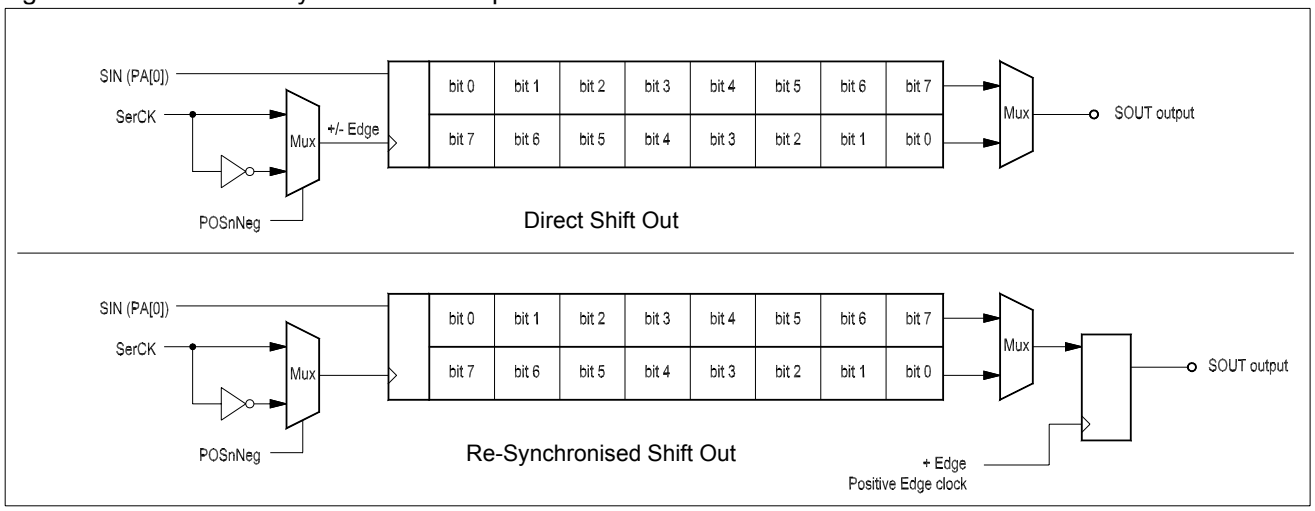
An interrupt request **IRQSerial** is generated after the eight shift operations are done. This signal is set by the last negative edge of the serial interface clock on PA[1] (master or slave mode) and is reset to 0 by the next write of **Start** or by any reset. This interrupt can be masked with register **RegIRQMask2.** For more details about the interrupt handling see chapter 11.

Serial data input on PA[0] is sampled by the positive or negative serial shifting clock edge, as selected by the Control Register **POSnNeg** bit**.** Serial data input is shifted in LSB first or MSB first, as selected by the Control Register **MSBnLSB** bit**.**

7.2.1 Output Modes

Serial data output is given out in two different ways. Refer also to Figures 15 and 16.

Figure 13. Direct or Re-Synchronized Output



- **OM[0]** = 0 :

The serial output data is generated with the selected shift register clock (**POSnNeg**). The first data bit is available directly after the **Start** bit is set.

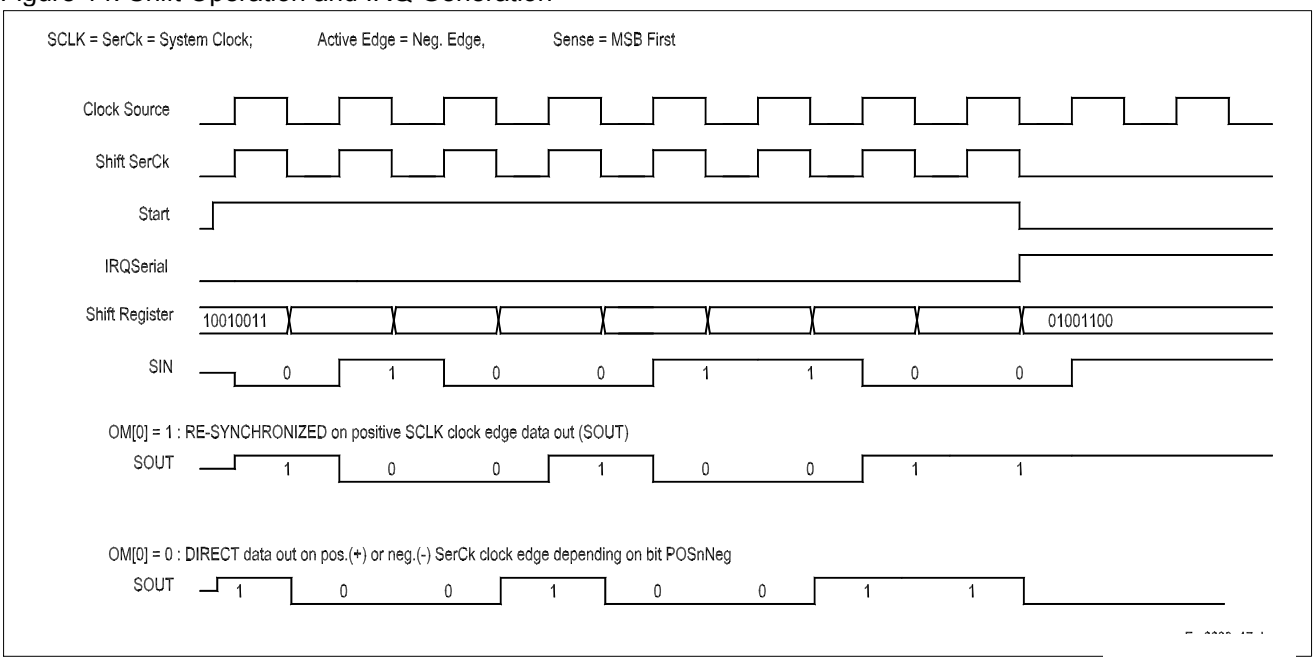
- **OM[0]** = 1 :

The serial output data is re-synchronized by the positive serial interface clock edge, independent of the selected clock shifting edge. The first data bit is available on the first positive serial interface clock edge after Start='1'.

Table 7.2.1 Output Mode Selection in **RegSCntl2**

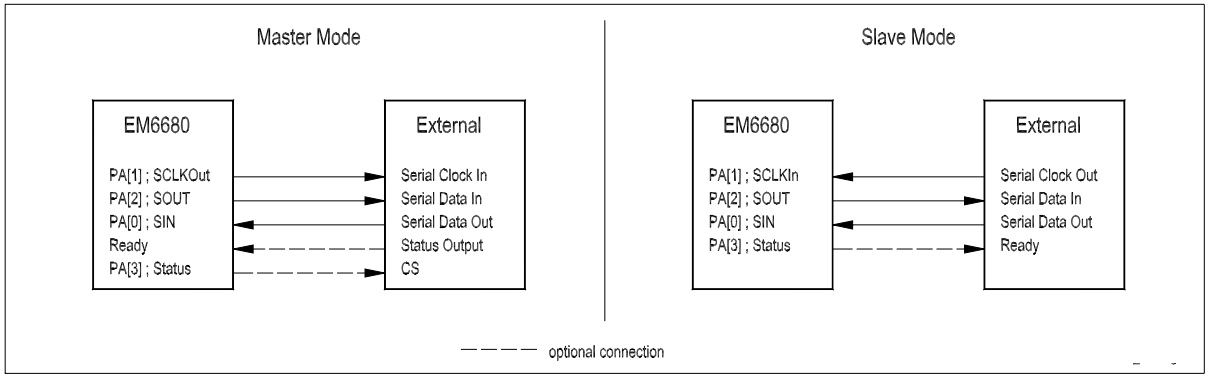
OM[0]	Output mode	Description
0	Serial-Direct	Direct shift pos. or neg. edge data out
1	Serial-Synchronized	Re-synchronized positive edge data shift out

Figure 14. Shift Operation and IRQ Generation



Note : A write operation in the control registers or in the data registers while **Start** is high will change internal values and may cause an error condition. The user must take care of the serial interface status before writing internal registers. In order to read the correct values on the data registers, the shift operation must be halted during the read accesses.

Figure 15. Example of Basic Serial Port Connections



7.3 Serial Interface Registers

Table 7.3.1 Register **RegSCntl1**

Bit	Name	Reset	R/W	Description
3	MS1	0	R/W	Frequency selection
2	MS0	0	R/W	Frequency selection
1	POSnNeg	0	R/W	Positive or negative clock edge selection for shift operation
0	MSBnLSB	0	R/W	Shift MSB or LSB value first (0=LSB first)

Default "0" is: Slave mode external clock, negative edge, LSB first

Table 7.3.2 Frequency and Master Slave Mode Selection

MS1	MS0	Description
0	**0**	Slave mode: Clock from external
0	**1**	Master mode Ck[14]: System clock / 4
1	**0**	Master mode Ck[15]: System clock / 2
1	**1**	Master modeCk[16]: System clock

Table 7.3.3 Register **RegSCntl2**

Bit	Name	Reset	R/W	Description
3	Start	0	R/W	Enabling the interface,
2	Status	0	R/W	Ready or Chip Select output on PA[3]
1	RCoscOff	0	R/W	RC oscill. disable when set @ ‘1’ if ExtCPUclkOn is ‘1’
0	OM[0]	0	R/W	‘0’: Direct shift Output, ‘1’: Output resynchronized

Default "0" is: Interface disabled, status 0, direct shift output.

Table 7.3.4 Register **RegSDataL**

Bit	Name	Reset	R/W	Description
3	SerDataL[3]	0	R/W	Serial data low nibble
2	SerDataL[2]	0	R/W	Serial data low nibble
1	SerDataL[1]	0	R/W	Serial data low nibble
0	SerDataL[0]	0	R/W	Serial data low nibble

Default "0" is: Data equal 0.

Table 7.3.5 Register **RegSDataH**

Bit	Name	Reset	R/W	Description
3	SerDataH[3]	0	R/W	Serial data high nibble
2	SerDataH[2]	0	R/W	Serial data high nibble
1	SerDataH[1]	0	R/W	Serial data high nibble
0	SerDataH[0]	0	R/W	Serial data high nibble

Default "0" is: Data equal 0.

8. 10-bit Counter

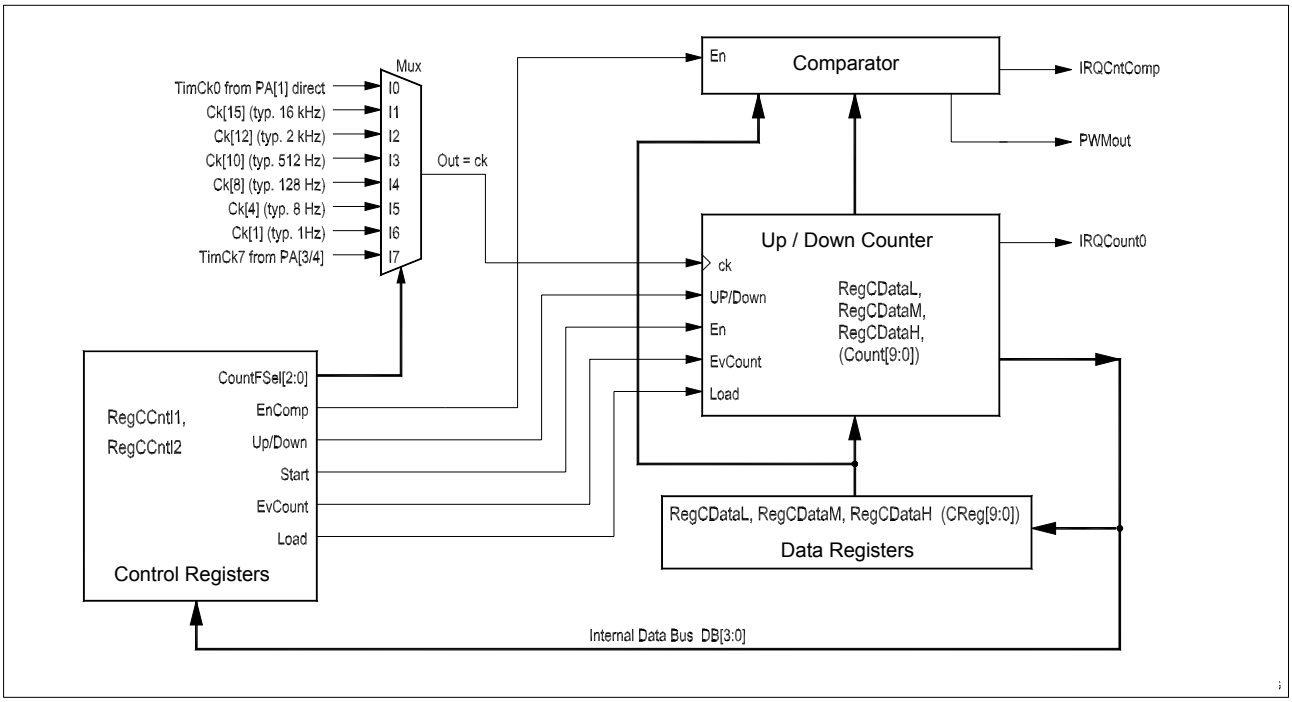
The EM6580 has a built-in universal cyclic counter. It can be configured as 10, 8, 6 or 4-bit counter. If 10-bits are selected we call that full bit counting, if 8, 6 or 4-bits are selected we call that limited bit counting.
The counter works in up- or down count mode. Eight clocks can be used as the input clock source, six of them are prescaler frequencies and two are coming from the input pads PA[1] (direct only) and PA[3/4] (direct or debounced). In this case the counter can be used as an event counter.

The counter generates an interrupt request **IRQCount0** every time it reaches 0 in down count mode or 3FF in up count mode. Another interrupt request **IRQCntComp** is generated in compare mode whenever the counter value matches the compare data register value. Each of this interrupt requests can be masked (default). See section 9 for more information about the interrupt handling.

A 10-bit data register **CReg[9:0]** is used to initialize the counter at a specific value (load into **Count[9:0]**). This data register (**CReg[9:0]**) is also used to compare its value against **Count[9:0]** for equivalence.

A Pulse-Width-Modulation signal (PWM) can be generated and output on port B terminal PA[0] or PA[1].

Figure 16. 10-bit Counter Block Diagram



8.1 Full and Limited Bit Counting

In Full Bit Counting mode the counter uses its maximum of 10-bits length (default). With the **BitSel[1,0]** bits in register **RegCDataH** one can lower the counter length, for IRQ generation, to 8, 6 or 4 bits. This means that actually the counter always uses all the 10-bits, but IRQCount0 generation is only performed on the number of selected bits. The unused counter bits may or may not be taken into account for the **IRQComp** generation depending on bit **SelIntFull**. Refer to chapter 8.4.

Table 8.1.1. Counter length selection

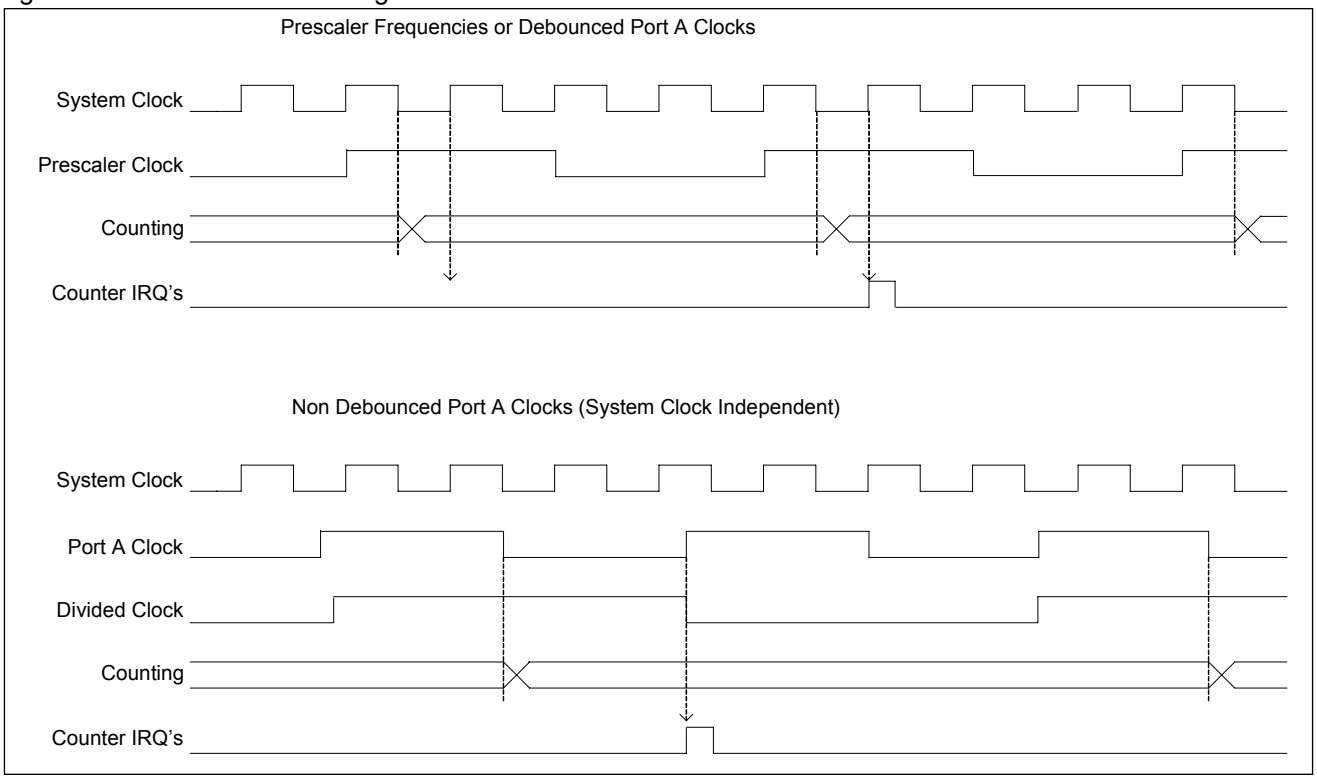
BitSel[1]	BitSel[0]	counter length
0	0	10-Bit
0	1	8-Bit
1	0	6-Bit
1	1	4-Bit

8.2 Frequency Select and Up/Down Counting

Eight (8) different input clocks can be selected to drive the Counter. The selection is done with bits **CountFSel2…0** in register **RegCCntl1**. Six (6) of this input clocks are coming from the prescaler. The maximum prescaler clock frequency for the counter is half the system clock SysClk and the lowest is 1Hz typ. Therefore a complete counter roll over can take as much as 17.07 minutes (1Hz clock, 10 bit length) or as little as 977 µs (Ck[15] typ 16.3kHz, 4 bit length). The **IRQCount0**, generated at each roll over, can be used for time bases, measurements length definitions, input polling, wake up from Halt mode, etc. The **IRQCount0** and **IRQComp** are generated with the system clock Ck[16] rising edge. IRQCount0 condition in up count mode is : reaching 3FF if 10-bit counter length (or FF, 3F, F in 8, 6, 4-bit counter length). In down count mode the condition is reaching '0'. The non-selected bits are 'don't care'. For IRQComp refer to section 8.4.

Note: The Prescaler and the Microprocessor clock's are usually non-synchronous, therefore time bases generated are max. n, min. n-1 clock cycles long (n being the selected counter start value in count down mode). However the prescaler clock can be synchronized with µP commands using for instance the prescaler reset function.

Figure 17. Counter Clock Timing



The two remaining clock sources are coming from the PA[1] or PA[3/4] terminals. Refer to Figure 10 on page 15 for details. Input PA[1] can be only direct non-debounce input, second PA[3/4] can be either debounce (Ck[11] or Ck[8]) or direct input, the input polarity can also be chosen. The outputs for Timer clock inputs are named TimCk0 and TimCk7 respectively. For the debouncer and input polarity selection refer to chapter 6.

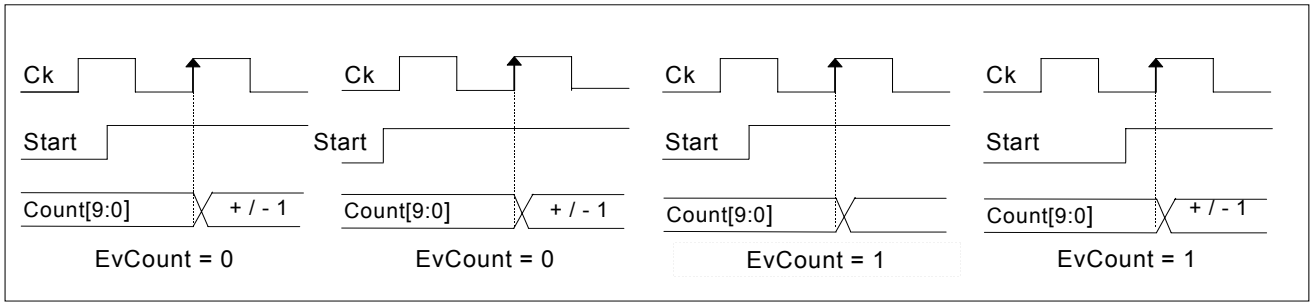
In the case of port A input clock without debouncer, the counting clock frequency will be half the input clock on port A. The counter advances on every odd numbered port A negative edge (divided clock is high level). **IRQCount0** and **IRQComp** will be generated on the rising PA[3/4] or PA[1] input clock edge. In this condition the EM6580 is able to count with a higher clock rate as the internal system clock (Hi-Frequency Input). Maximum port A input frequency is limited to 500kHz (@$V_{dd} \geq 1.5$ V). If higher frequencies are needed, please contact EM Microelectronic's.

In both, up or down count (default) mode, the counter is cyclic. The counting direction is chosen in register **RegCCntl1** bit **Up/Down** (default '0' is down count). The counter increases or decreases its value with each positive clock edge of the selected input clock source. Start up synchronization is necessary because one can not always know the clock status when enabling the counter. With EvCount=0, the counter will only start on the next positive clock edge after a previously latched negative edge, while the **Start** bit was already set to '1'. This synchronization is done differently if event count mode (bit **EvCount**) is chosen. Refer also to Figure 18. Internal Clock Synchronization.

8.3 Event Counting

The counter can be used in a special event count mode where a certain number of events (clocks) on the PA[1] (only non-debounced and only rising edge) or PA[3/4] input are counted. In this mode the counting will start directly on the next active clock edge on the selected port A input.

Figure 18. Internal Clock Synchronization



The Event Count mode is switched on by setting bit **EvCount** in the register **RegCCntl2** to '1'. PA[3] or PA[4] input depending on **IrqPA[3l/4h]** bit in **RegPaCntl1** can be inverted depending on **edgeFallingPA[3/4]** in register **RegPaCntl1** and should be debounced. The debouncer is switched on with **debounceNoPA[3/4]** at '0' in the same register. Its frequency depends on the bit **DebSel** from register **RegPresc** setting. Refer also to Figure 10 for PortA Inputs Function. As already said for other PA[1] input only possibility is to count rising non-debounced edges.

A previously loaded register value (**CReg[9:0]**) can be compared against the actual counter value (**Count[9:0]**). If the two are matching (equality) then an interrupt (**IRQComp**) is generated. The compare function is switched on with the bit **EnComp** in the register **RegCCntl2**. With **EnComp** = 0 no **IRQComp** is generated. Starting the counter with the same value as the compare register is possible, no IRQ is generated on start. Full or Limited bit compare are possible, defined by bit **SelIntFull** in register **RegSysCntl1**.
EnComp must be written after a load operation (**Load** = 1). Every load operation resets the bit EnComp.

Full bit compare function.

Bit **SelIntFull** is set to '1'. The function behaves as described above independent of the selected counter length. Limited bit counting together with full bit compare can be used to generate a certain amount of IRQCount0 interrupts until the counter generates the **IRQComp** interrupt. With ***PWMOn***='1' the counter would have automatically stopped after the **IRQComp**, with ***PWMOn***='0' it will continue until the software stops it. **EnComp** must be cleared before setting **SelIntFull** and before starting the counter again. Be careful, **PWMoutPA[0]** also redefines the port PA[0] or **PWMoutPA[1]** the PA[1] output data. (refer to section 8.4).
The signal PWMOn is acombination of **PWMOutPA[0], PWMOutPA[1], SerialCkPA[1]**

PWMOn = (PWMOutPA[0] **OR** PWMOutPA[1]) **AND NOT**(SeriaCktPA[1]))

Limited bit compare

With the bit **SelIntFull** set to '0' (default) the compare function will only take as many bits into account as defined by the counter length selection **BitSel[1:0]** (see chapter 6.3).

8.4 Pulse Width Modulation (PWM)

The PWM generator uses the behavior of the Compare function (see above) so **EnComp** must be set to activate the PWM function.. At each Roll Over or Compare Match the PWM state - which is output on port PA[0] or PA[1] - will toggle. The start value on PA[0] or PA[1] is forced while **EnComp** is 0 the value is depending on the up or down count mode. Every counter value load operation resets the bit **EnComp** and therefore the PWM start value is reinstalled.

One can output PWM signal to PA[0] or PA[1]. Setting **PWMoutPA[0]** to '1' in register **RegPaCntl3** routes the counter PWM output to PA[0]. Insure that PA[0] is set to output mode. Setting **PWMoutPA[1]** to '1' in register **RegPaCntl3** routes the counter PWM output to PA[1]. Insure that PA[1] is set to output mode. Refer to section 6.3 and 6.4 for the port A output setup.

The PWM signal generation is independent of the limited or full bit compare selection bit **SelIntFull.** However if **SelIntFull** = 1 (FULL) and the counter compare function is limited to lower than 10 bits one can generate a predefined number of output pulses. In this case, the number of output pulses is defined by the value of the unused counter bits. It will count from the start value until the IRQComp match.

One must not use a compare value of hex 0 in up count mode nor a value of hex 3FF (or FF,3F, F if limited bit compare) in down count mode.

For instance, loading the counter in up count mode with hex 000 and the comparator with hex C52 which will be identified as :

- bits[11:10] are limiting the counter to limits to 4 bits length, =03	(BitSel[1,0])
- bits [9:4] are the unused counter bits = hex 05 (bin 000101),	(number of PWM pulses)
- bits [3:0] (comparator value = 2).	(length of PWM pulse)

Thus after 5 PWM-pulses of 2 clocks cycles length the Counter generates an **IRQComp** and stops.
The same example with SelIntFull=0 (limited bit compare) will produce an unlimited number of PWM at a length of 2 clock cycles.

8.4.1 How the PWM Generator works.

For Up Count Mode; Setting the counter in up count and PWM mode the PA[0] or PA[1] PWM output is defined to be 0 (**EnComp**=0 forces the PWM output to 0 in upcount mode, 1 in downcount). Each Roll Over will set the output to '1' and each Compare Match will set it back to '0'. The Compare Match for PWM always only works on the defined counter length. This, independent of the **SelIntFull** setting which is valid only for the IRQ generation. Refer also to the compare setup in chapter 0.

In above example the PWM starts counting up on hex 0,
2 cycles later compare match → PWM to '0',
14 cycles later roll over → PWM to '1'
2 cycles later compare match → PWM to '0' , etc. until the completion of the 5 pulses.

The normal IRQ generation remains on during PWM output. If no IRQ's are wanted, the corresponding masks need to be set.

Figure 19. PWM Output in Up Count Mode

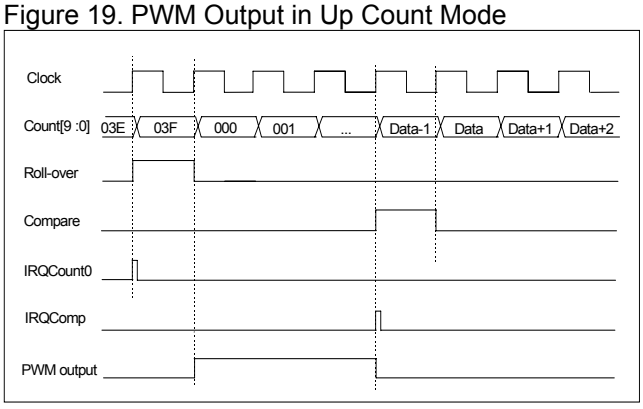
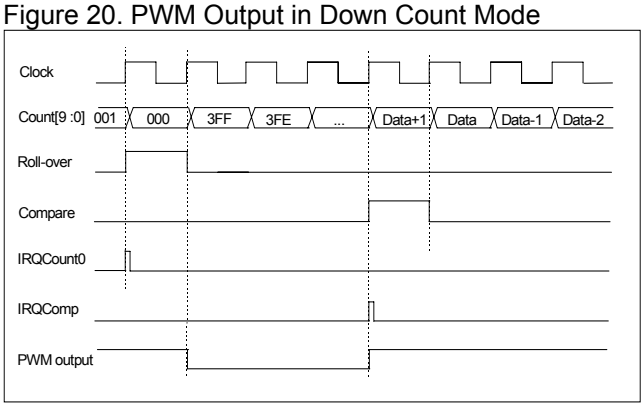


Figure 20. PWM Output in Down Count Mode



In Down Count Mode everything is inverted. The PWM output starts with the '1' value. Each Roll Over will set the output to '0' and each Compare Match will set it back to '1'. Due to this positive pulse length is always longer by 1 selected clock period compared to written value. Example: for 25% positive pulse duty cycle on 4 bit counter one must write 3 in counter instead of 4.
For limited pulse generation one must load the complementary pulse number value. I.e. for 5 pulses counting on 4 bits load bits[9 :4] with hex 3A (bin 111010).

8.4.2 PWM Characteristics

PWM resolution is	: 10bits (1024 steps), 8bits (256 steps), 6bits (64 steps) or 4 bits (16 steps)			
the minimal signal period is	: 16 (4-bit) x Fmax*	→ 16 x 1/Ck[15]	→ 977 µs	(32 KHz)
the maximum signal period is	: 1024 x Fmin*	→ 1024 x 1/Ck[1]	→ 1024 s	(32 KHz)
the minimal pulse width is	: 1 bit	→ 1 x 1/Ck[15]	→ 61 µs	(32 KHz)

* This values are for Fmax or Fmin derived from the internal system clock (32kHz). Much shorter (and longer) PWM pulses can be achieved by using the port A as frequency input.

One must not use a compare value of hex 0 in up count mode nor a value of hex 3FF (or FF,3F, F if limited bit compare) in down-count mode.

8.5 Counter Setup

RegCDataL[3:0], RegCDataM[3:0], RegCDataH[1:0] are used to store the initial count value called $\mathbf{C_{Reg}[9:0]}$ which is written into the count register bits **Count[9:0]** when writing the bit **Load** to '1' in **RegCCntl2**. This bit is automatically reset thereafter. The counter value **Count[9:0]** can be read out at any time, except when using non-debounced high frequency port A input clock. To maintain data integrity the lower nibble **Count[3:0]** must always be read first. The **ShCount[9:4]** values are shadow registers to the counter. To keep the data integrity during a counter read operation (3 reads), the counter values [9:4] are copied into these registers with the read of the count[3:0] register. If using non-debounced high frequency port A input the counter must be stopped while reading the **Count[3:0]** value to maintain the data integrity.
In down count mode an interrupt request **IRQCount0** is generated when the counter reaches 0. In up count mode, an interrupt request is generated when the counter reaches 3FF (or FF,3F,F if limited bit counting).

Never an interrupt request is generated by loading a value into the counter register.

When the counter is programmed from up into down mode or vice versa, the counter value **Count[9:0]** gets inverted. As a consequence, the initial value of the counter must be programmed after the **Up/Down** selection.

Loading the counter with hex 000 is equivalent to writing stop mode, the **Start** bit is reset, no interrupt request is generated.

How to use the counter;

If PWM output is required one has to decide first on which PA port to put it. After corresponding port Output Enable **OEnPA[n]** must be set **PWMoutPA[n]** = 1 in step 5. (n= 0 or 1)

1st, set the counter into stop mode (**Start**=0).
2nd, select the frequency and up- or down count mode in **RegCCntl1.**
3rd, write the data registers **RegCDataL, RegCDataM, RegCDataH** (counter start value and length)
4th, load the counter, **Load**=1, and choose the mode. (**EvCount**, **EnComp**=0)
5th, select bits **PWMoutPA[n]** in **RegPaCntl3** and **SelIntFull** in **RegSysCntl1**
6th, if compare mode desired , then write **RegCDataL, RegCDataM, RegCDataH** (compare value)
7th, set bit **Start** and select **EnComp** in **RegCCntl2**.

8.6 10-bit Counter Registers

Table 8.6.1. Register **RegCCntl1**

Bit	Name	Reset	R/W	Description
3	Up/Down	0	R/W	Up or down counting
2	CountFSel2	0	R/W	Input clock selection
1	CountFSel1	0	R/W	Input clock selection
0	CountFsel0	0	R/W	Input clock selection

Default : PA0 ,selected as input clock, Down counting

Table 8.6.2. Counter Input Frequency Selection with **CountFSel[2..0]**

CountFSel2	CountFSel1	CountFSel0	clock source selection
0	0	0	Port A PA[1] = non debounced only TimCk0
0	0	1	Prescaler Ck[15] typ. 16 kHz
0	1	0	Prescaler Ck[12] typ. 2 kHz
0	1	1	Prescaler Ck[10] typ. 512 Hz
1	0	0	Prescaler Ck[8] typ. 128 Hz
1	0	1	Prescaler Ck[4] typ. 8 Hz
1	1	0	Prescaler Ck[1] typ. 1 Hz
1	1	1	Port A PA[3/4]

Table 8.6.3. Register **RegCCntl2**

Bit	Name	Reset	R/W	Description
3	Start	0	R/W	Start/Stop control
2	EvCount	0	R/W	Event counter enable
1	EnComp	0	R/W	Enable comparator
0	Load	0	R/W	Write: load counter register; Read: always 0

Default : Stop, no event count, no comparator, no load

Table 8.6.4. System Control register **RegSysCntl1**

Bit	Name	Reset	R/W	Description
3	IntEn	0	R/W	*General interrupt enable*
2	Sleep	0	R/W	*Sleep mode*
1	SetIntFull	0	R/W	Compare Interrupt select (**note** 1)
0	***ChTmDis***	p 0*	R/W	Disable Test modes by setting it to 1 **(MUST be DONE)**

p 0* ***ChTmDis*** is cleared on POR to be able to enter test modes at EM. **One of first instructions not to enter test mode by mistake even if protocol to do it is difficult to achive is to set this bit to 1.** Because circuit starts itself with its ROM code also on tester user has to respect minimal number of instructions before setting this bit to be able to test the circuit on tester.

Table 8.6.5. Number of instructions before cutting Test access

CPU frequency	Min Nb. of instructions before ChTmDis is set or PortPA[0] declared as an output
Basic frequency (32 kHz or 50 kHz)	4
Basic f. x 2 (64 kHz or 100 kHz)	8
Basic f. x 4 (128 kHz or 200 kHz)	16
Basic f. x 8 (256 kHz or 400 kHz)	32
Basic f. x 16 (500 kHz or 800 kHz)	64

By writing to **RegSysCntl1** – setting ChTmDis to 1 PORstatus will be cleared.
Test mode is totally disabled also if PortPA[0] is declared as an output. OenPA[0] = '1'

(**note** 1) Default : Interrupt on limited bit compare for Counter

Table 8.6.6. Register **RegCDataL**, Counter/Compare Low Data Nibble

Bit	Name	Reset	R/W	Description
3	CReg[3]	0	W	Counter data bit 3
2	CReg[2]	0	W	Counter data bit 2
1	CReg[1]	0	W	Counter data bit 1
0	CReg[0]	0	W	Counter data bit 0
3	Count[3]	0	R	Data register bit 3
2	Count[2]	0	R	Data register bit 2
1	Count[1]	0	R	Data register bit 1
0	Count[0]	0	R	Data register bit 0

Table 8.6.7. Register **RegCDataM**, Counter/Compare Middle Data Nibble

Bit	Name	Reset	R/W	Description
3	CReg[7]	0	W	Counter data bit 7
2	CReg[6]	0	W	Counter data bit 6
1	CReg[5]	0	W	Counter data bit 5
0	CReg[4]	0	W	Counter data bit 4
3	ShCount[7]	0	R	Data register bit 7
2	ShCount[6]	0	R	Data register bit 6
1	ShCount[5]	0	R	Data register bit 5
0	ShCount[4]	0	R	Data register bit 4

Table 8.6.8. Register **RegCDataH**, Counter/Compare High Data Nibble

Bit	Name	Reset	R/W	Description
3	BitSel[1]	0	R/W	Bit select for limited bit count/compare
2	BitSel[0]	0	R/W	Bit select for limited bit count/compare
1	CReg[9]	0	W	Counter data bit 9
0	CReg[8]	0	W	Counter data bit 8
1	ShCount[9]	0	R	Data register bit 9
0	ShCount[8]	0	R	Data register bit 8

Table 8.6.9. Counter Length Selection

	BitSel[1]	BitSel[0]	counter length
	0	0	10-Bit
	0	1	8-Bit
	1	0	6-Bit
	1	1	4-Bit

9. Supply Voltage Level Detector / 4-bit ADC

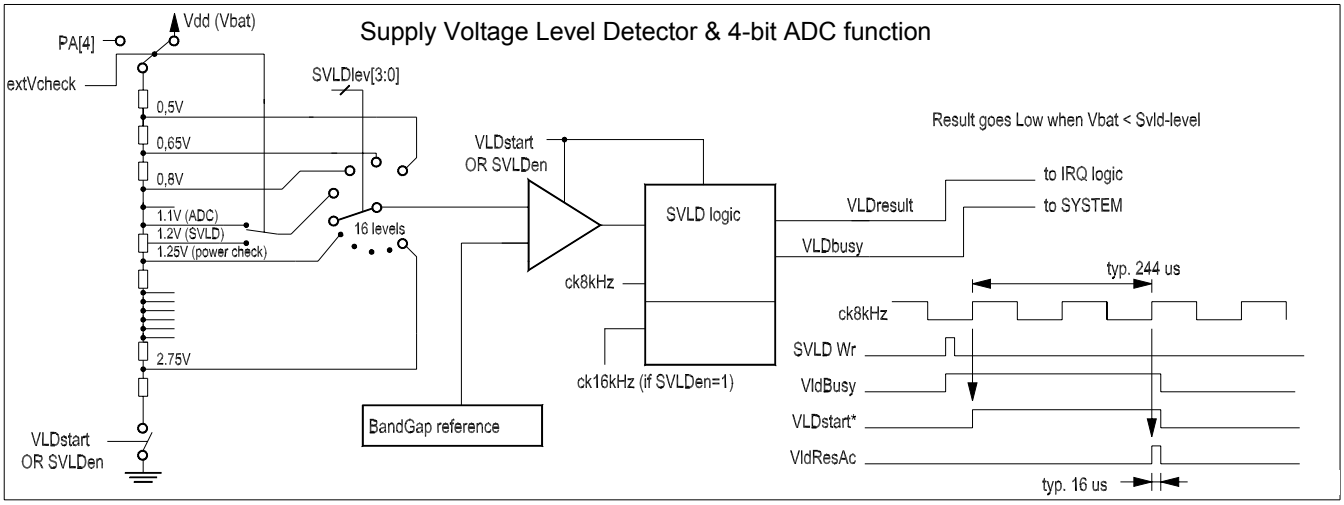
The EM6580 has a built-in circuitry made up of a comparator with band-gap reference and a resistor divider chain with 16 terminals to detect levels from 0.5V to 2.75V with a step of 150mV. This can be used as:

a. **S**upply **V**oltage **L**evel **D**etector (SVLD) to compare the positive power supply level V_{dd} (V_{bat}) against levels which are in the range of V_{ddmin} to V_{ddmax}. There are 12 pre-selected levels in the range from 1.2 to 2.75V. In this case **ExtVcheck** must be cleared to '0' (default).

b. Simple 4-bit **A**nalogue to **D**igital **C**onverter – ADC. Setting the **ExtVcheck** bit to '1' makes the PA[4] input an analog ADC input. PA[4] input voltage must not exceed V_{bat} + 0.3V

In Sleep mode both functions are disabled.

Figure 21. SVLD / 4-bit ADC schematic with controls and timing



SVLD level5 (1.25V) or SVLD level9 (1.85V) is used during Power On Reset for Power-Check to check the minimum operating voltage before the POR signal is released as described in Chapter 4.1.

When used as a SVLD the **ExtVCheck** bit in register **RegVldCntl** must be cleared to '0'. Then V_{dd} (V_{bat}) is selected as the input to the resistive divider which provides the comparator inputs. The SVLD level must be selected by writing the **RegSVLDlev** register. For proper operation only levels from #4 to #15 can be selected. Then the CPU activates the voltage comparison by writing the **VLDstart** bit to '1' in the register **RegVLDCntl**. The actual measurement starts on the next ck[14] (8kHz @ 32kHz SysClk) falling edge and lasts typ. 260 us. The busy flag **VldBusy** stays high from the time **VLDStart** is set to '1' until the measurement is finished. The worst case time until the result is available is 3.125 * ck[14] prescaler clock periods (32kHz → 382us). See figure 24 for details.

During the actual measurement (typ. 260us) the device will draw typically an additional 4µA of I_{VDD} current @ V_{dd}=1.5V. After the end of the measurement an interrupt request **IRQSvld** can be generated if V_{dd} is lower than the level which was selected. The interrupt is generated only if the **MaskIRQSvld** bit is set to '1'. The result is available by inspection of the bit **VLDResult**. If the result is '0', then the power supply voltage was lower than the detection level value. If '1' the power supply voltage was higher than the detection level value. The value of **VLDResult** is not guaranteed while **VldBusy=1**.

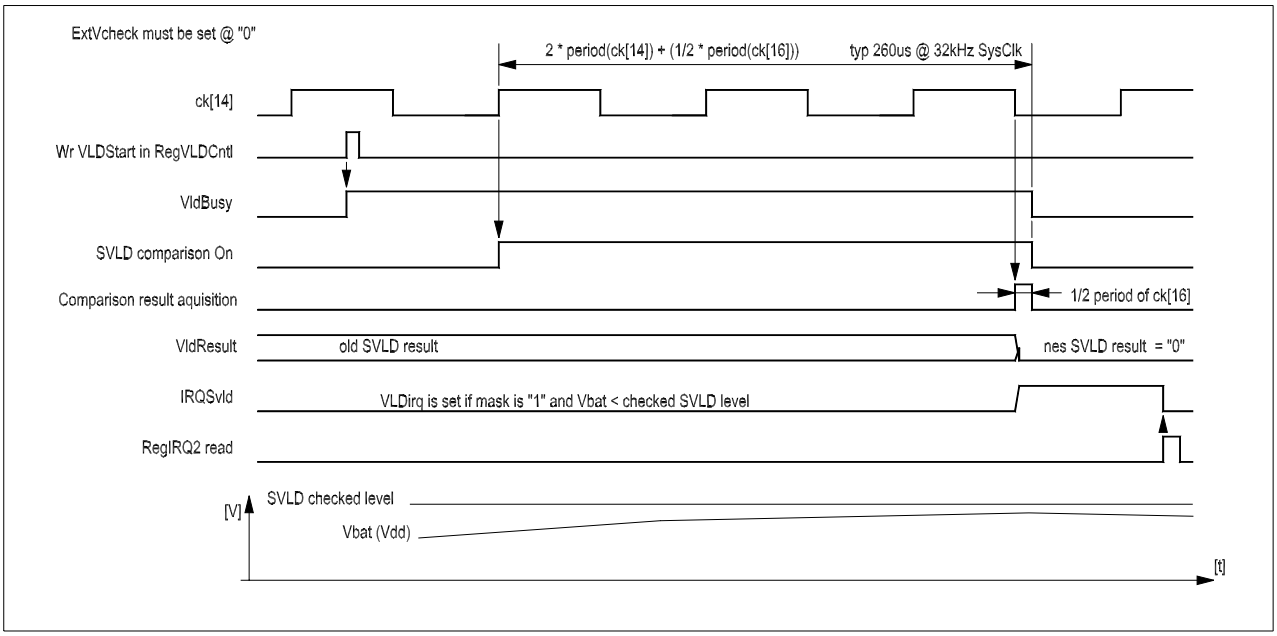
An interrupt can be generated only if V_{dd}(V_{bat}) is lower than the selected level. **IRQSvld** bit is cleared by reading **RegIRQ2**.

Table 9.1 register **RegVldCntl**

Bit	Name	Reset	R/W	Description
3	ExtVcheck	0	W	PA[4] as positive input of divider chain
3	VLDResult	0	R*	VLD result flag
2	VLDStart	0	W	VLD start command
2	VLDBusy	0	R	VLD busy flag is on Until compare is finished
1	SVLDen	0	R/W	SVLD comparator is On continuously
0	NoWDtim	p	R/W	No watchdog timer

R*; VLDResult is not guaranteed while VLDBusy=1

Figure 22. SVLD timing



The **VLDresult** bit from the previous measurement stays in the register until the new measurements is finished. For good measurements external noise or CPU activity should be as low as possible during the comparison.

Table 9.2 register **RegSVLDlev**

Bit	Name	POR	R/W	Description
3	SVLDlev[3]	0	R/W	SVLD level select bit #3
2	SVLDlev[2]	1	R/W	SVLD level select bit #2
1	SVLDlev[1]	0	R/W	SVLD level select bit #1
0	SVLDlev[0]	1	R/W	SVLD level select bit #0

Table 9.3 SVLD level selection (typical values **V*SVLD*Nom**)

SVLDlev[3:0]				Nb.:	ADC	SVLD	Description of specialitis
MSB			LSB	Level #	ADCNom	SVLDNom	
0	0	0	0	0	0.5 V	Do not set	Do not use this level with SVLD
0	0	0	1	1	0.65 V	Do not set	Do not use this level with SVLD
0	0	1	0	2	0.80 V	Do not set	Do not use this level with SVLD
0	0	1	1	3	0.95 V	Do not set	Do not use this level with SVLD
0	1	0	0	4	1.10 V	Do not set	Do not use this level with SVLD
0	1	0	1	5	1.25 V	Do not set	Do not use this level with SVLD
0	1	1	0	6	1.40 V	Do not set	Do not use this level with SVLD
0	1	1	1	7	1.55 V	Do not set	Do not use this level with SVLD
1	0	0	0	8	1.70 V	Do not set	Do not use this level with SVLD
1	0	0	1	9	1.85 V	Do not set	Do not use this level with SVLD
1	0	1	0	10	2.00 V	2.00 V	Default level after POR for Power Check
1	0	1	1	11	2.15 V	2.15 V	
1	1	0	0	12	2.30 V	2.30 V	
1	1	0	1	13	2.45 V	2.45 V	
1	1	1	0	14	2.60 V	2.60 V	
1	1	1	1	15	2.75 V	2.75 V	

External source is coming from PA[4] as explained in Chapter 2 and shown on figure 10.

To implement a 4-bit ADC first **ExtVcheck** bit must be set to '1', that PA[4] input is connected to positive side of resistor divider chain. To find the level as fast as possible with successive approximation for instance it is advised to Set **SVLDen** bit to '1' to have comparator and resistive divider chain operational all the time when the PA[4] input is sampled with ck[15] (typ. 16kHz @ 32kHz SysClk) frequency. With SVLDlev[3:0] we can select one of 16 possible levels to check and by making max. 4 measurements at 4 different levels (if input PA[4] is stable) we have the result with successive approximation method.

In this case PA[4] input is blocked for all other functions, because its level can be in a zone where logic ‘0’ or ‘1’ are not well defined and this would generate an over consumption otherwise. So it is dedicated only to SVLD comparator input to be compared with internal band-gap reference. NoPullPA[4] must be set to ‘1’ - Pull-UP/Down must be removed by register also.

In both cases if V_{bat} (V_{dd}) or PA[4] level is tested than if selected tested level lower an IRQ can be generated if enabled.

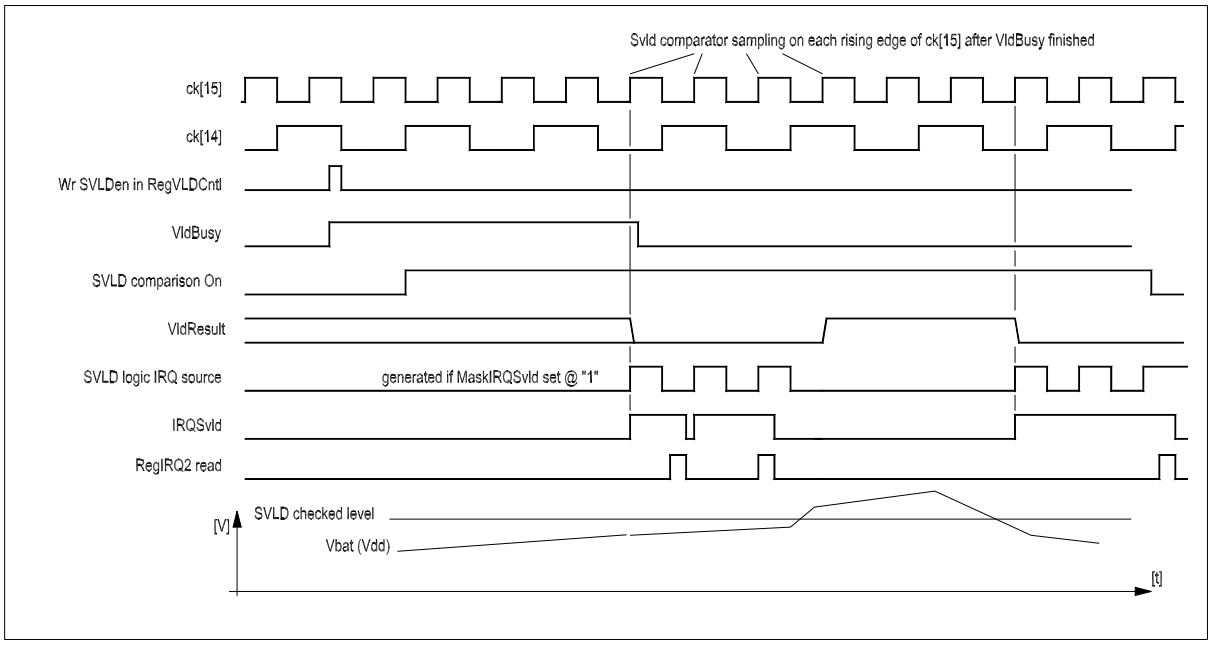
With **SVLDen** bit one can switch on the band-gap, resistive divider and Comparator continuously. Like that one can monitor V_{bat} or PA[4] level continuously, at higher frequency (ck[15]). Only at the beginning after setting the **SVLDen** at ‘1’ one has to wait until **VLDbusy** drops to ‘0’ indicating that system is powered up (band-gap reference and resistor divider are stabilized and comparator is ready to give proper result). This will increase power consumption by typ. 4µA @ V_{dd}=1.5V while used.

During continuously monitoring one can change **RegSVLDlev** register value on fly and the new result should be read only after about 1.5 * ck[15] to be sure it is a result of a new **SVLDlev** selection. Depending on CPUclk and divisions to obtained SysClk this can be 2 / 6 / 10 / 20 / 36 instruction after **RegSVLDlev** change for multiples by 1 / 2 / 4 / 8 / 16.

When fast monitoring is not necessary any more one can remove it by clearing **SVLDen** to ‘0’.

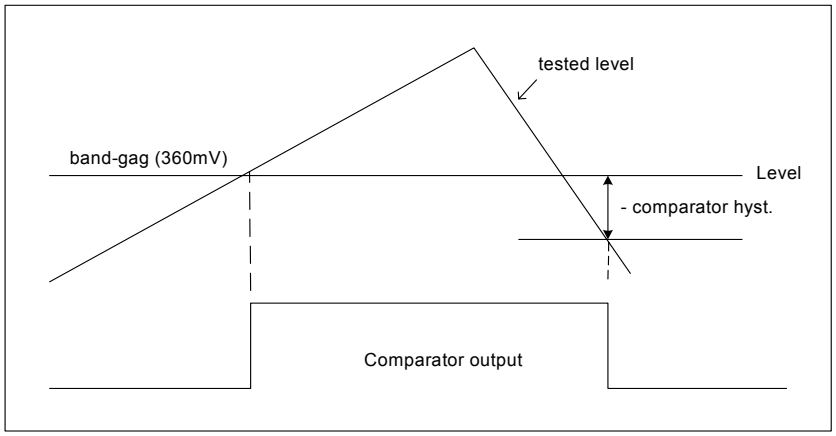
When SVLD logic is used for this fast monitoring IRQ can also be generated when checked level falls below its value.

Figure 23. SVLD timing in “ADC” mode when SVLDen set @ “1”



Due to IRQSvld which can come very fast – with ck[15] there is danger that immediately after coming out from IRQ subroutine new IRQSvld which came during that time put uC back in IRQ subroutine and software can be stacked at this place until checked input is lower then SVLD level. Otherwise IntEn register must be cleared in IRQ subroutine already !! or even better to use this function by reading the SVLD result only and not setting the MaskIRQSvld.

10. ADC/SVLD Comparator characteristics



Comparator hysteresis is adaptive in 4 steps controlled by MSB of RegSVLDlev register to cover all tested levels from 0.5V to 3.0V to give a typ. “tested level” hysteresis of 40mV.

Negative hysteresis on comparator is implemented to eliminate comparator output oscillation arround switching point.

11. RAM

The EM6580 has one 80x4 bit static RAM built-in located on addresses hex 0 to 4F. All the RAM nibbles are direct addressable.

Figure 24. RAM Architecture

	RAM 80 x 4 = direct addressable	
Adr [hex]	**RAM location**	**Read / Write**
4F	RAM_79	4 bit R/W
4E	RAM_78	4 bit R/W
4D	RAM_77	4 bit R/W
4B	RAM_76	4 bit R/W
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
04	RAM_04	4 bit R/W
03	RAM_03	4 bit R/W
02	RAM_02	4 bit R/W
01	RAM_01	4 bit R/W
00	RAM_00	4 bit R/W

RAM Extension : Unused R/W Registers can often be used as possible RAM extension. Be careful not to use registers which start, stop, or reset some functions.

12. Interrupt Controller

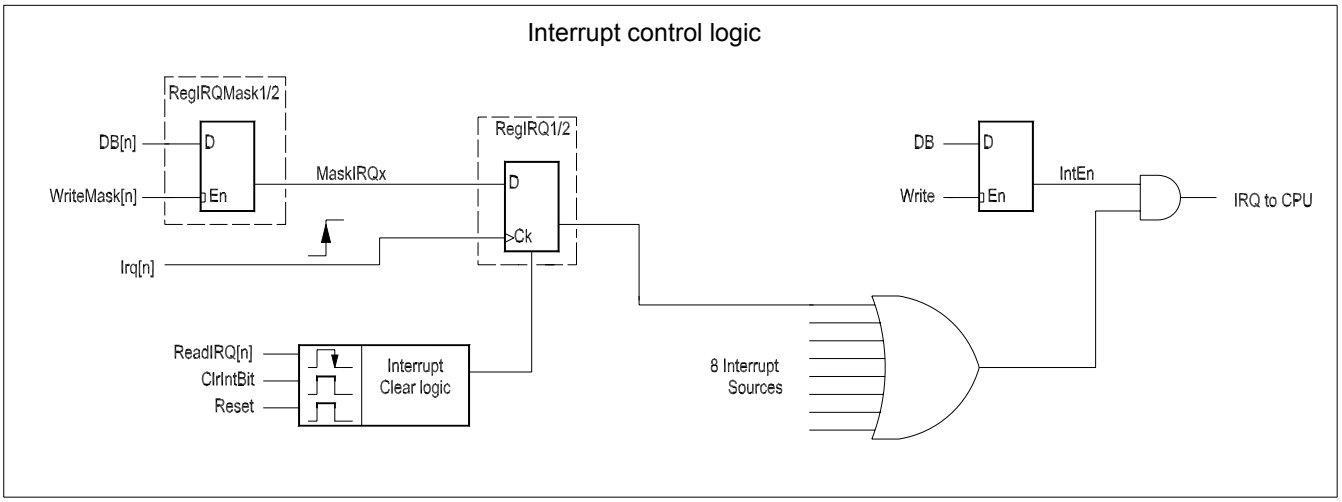
The EM6580 has 8 different interrupt request sources masked individually. These are:

External (2)	**- Port A,**	**PA[0/5] and PA[3/4] inputs**
	- Compare	**PA[4] input**
Internal (6)	**- Prescaler (2x)**	**ck[1], 64Hz/8Hz**
	- 10-bit Counter (2x)	**Count0, CountComp**
	- SVLD (1)	**End of measure when level is low**
	- Serial Interface (1)	**8 bit transfered**

The SVLD and the PA[4] level check share the same interrupt line.
Serial Interface could be put under Internal when Serial clock is coming form EM6580 or External when Serial clock is external.

To be able to send an interrupt to the CPU, at least one of the interrupt request flags must be set (**IRQxx**) and the general interrupt enable bit **IntEn** located in the register **RegSysCntl1** must be set to 1. The interrupt request flags can only be set by a positive edge of **IRQxx** with the corresponding mask register bit (**MaskIRQxx**) set to 1.

Figure 25. Interrupt control logic for generating and clearing interrupts



At power on or after any reset all interrupt request mask registers are cleared and therefore do not allow any interrupt request to be stored. Also the general interrupt enable **IntEn** is set to 0 (No IRQ to CPU) by reset.

After each read operation on the interrupt request registers **RegIRQ1** or **RegIRQ2** the contents of the addressed register are reset. Therefore one has to make a copy of the interrupt request register if there was more than one interrupt to treat. Each interrupt request flag may also be reset individually by writing 1 into it (ClrIntBit).

Interrupt handling priority must be resolved through software by deciding which register and which flag inside the register need to be serviced first.

Since the CPU has only one interrupt subroutine and because the **IRQxx** registers are cleared after reading, the CPU does not miss any interrupt request which comes during the interrupt service routine. If any occurs during this time a new interrupt will be generated as soon as the software comes out of the current interrupt subroutine.

Any interrupt request sent by a periphery cell while the corresponding mask is not set will not be stored in the interrupt request register. All interrupt requests are stored in their **IRQxx** registers depending only on their corresponding mask setting and not on the general interrupt enable status. Whenever the EM6580 goes into HALT Mode the **IntEn** bit is automatically set to 1, thus allowing to resume from Halt Mode with an interrupt.

12.1 Interrupt control registers

Table 12.1.1 Register **RegIRQ1**

Bit	Name	Reset	R/W	Description
3	IRQCount0	0	R/W*	Counter interrupt request when at 0
2	IRQCntComp	0	R/W*	Counter interrupt request when compare True
1	IRQPA[3/4]	0	R/W*	Port A PA[3/4] interrupt request
0	IRQPA[0/5]	0	R/W*	Port A PA[0/5] interrupt request

W*; Writing of 1 clears the corresponding bit.

Table 12.1.2 Register **RegIRQ2**

Bit	Name	Reset	R/W	Description
3	IRQHz1	0	R/W*	Prescaler interrupt request of 1Hz
2	IRQHz64/8	0	R/W*	Prescaler interrupt request of 64 Hz or 8 Hz
1	IRQSvld	0	R/W*	SVLD or Compare interrupt request
0	IRQSerial	0	R/W*	Serial interface interrupt request

W*; Writing of 1 clears the corresponding bit.

Table 12.1.3 Register **RegIRQMask1**

Bit	Name	Reset	R/W	Description
3	MaskIRQCount0	0	R/W	Counter when at 0 interrupt mask
2	MaskIRQCntComp	0	R/W	Counter compare True interrupt mask
1	MaskIRQPA[3/4]	0	R/W	Port A PA[3/4] interrupt mask
0	MaskIRQPA[0/5]	0	R/W	Port A PA[0/5] interrupt mask

Interrupt is not stored if the mask bit is 0.

Table 12.1.4 Register **RegIRQMask2**

Bit	Name	Reset	R/W	Description
3	MaskIRQHz1	0	R/W	Prescaler 1Hz interrupt mask
2	MaskIRQHz64/8	0	R/W	Prescaler 64 Hz or 8 Hz interrupt mask
1	MaskIRQSvld	0	R/W	SVLD or Compare interrupt mask
0	MaskIRQSerial	0	R/W	Serial interface interrupt mask

Interrupt is not stored if the mask bit is 0.

13. PERIPHERAL MEMORY MAP

Reset values are valid after power up or after every system reset.

Register Name	Add Hex	Add Dec.	Reset Value	Read Bits	Write Bits	Remarks
			b'3210	Read / Write Bits		
Ram1_0	00	0	xxxx	0: Data0 1: Data1 2: Data2 3: Data3		Direct addressable Ram 80 x 4 bit
Ram1_63	4F	79	xxxx	0: Data0 1: Data1 2: Data2 3: Data3		Direct addressable Ram 80 x 4 bit
RegPA0	50	80	0000	0: PA[0] 1: PA[1] 2: PA[2] 3: PA[3]	0: PAout[0] 1: PAout[1] 2: PAout[2] 3: PAout[3]	PortA [3:0] Direct input read, Output data register
RegPa0OE	51	81	0***0***00 ***0*** = after PORend	0: OEnPA[0] 1: OEnPA[1] 2: OEnPA[2] 3: OEnPA[3]		PortA [3:0] Output enable active Hi,
RegPaCntl1	52	82	pppp p = POR	0: EdgeFallingPA[0/5] 1: EdgeFallingPA[3/4] 2: debunceNoPA[0/5] 3: debunceNoPA[3/4]		PortA [3:0] control1 Debounce Yes/No & Faling / Rising edge
RegPaCntl2	53	83	pppp p = POR	0: IrqPA[0l/5h] 1: IrqPA[3l/4h] 2: WUchEnPA[0/5] 3: WUchEnPA[3/4]		PortA [3:0] control2 WakeUp on change enable & Irq source from PA select
Pa0noPDown	54	84	pppp p = POR	0: NoPdPA[0] 1: NoPdPA[1] 2: NoPdPA[2] 3: NoPdPA[3]		Option register Pull/down selection on PA[3:0] Default : pull-down ON
Pa0NchOpenDr	55	85	pppp p = POR	0: NchOpDrPA[0] 1: NchOpDrPA[1] 2: NchOpDrPA[2] 3: NchOpDrPA[3]		Option register N/channel Open Drain Output on PA[3:0] Default : CMOS output
RegFreqRst	56	86	ppxx	0: foutSel[0] 1: foutSel[1] 2: PA[3/4]resIn 3: InResAH		Output Frequency select and Input reset Control
RegSCntl1	57	87	0000	0: MSBnLSB 1: POSnNeg 2: MS0 3: MS1		Serial interface control 1
RegSCntl2	58	88	0000	0: OM[0] 1: RCoscOff 2: Status 3: Start		Serial interface control 2
RegSDataL	59	89	0000	0: SerDataL[0] 1: SerDataL[1] 2: SerDataL[2] 3: SerDataL[3]		Serial interface low data nibble
RegSDataH	5A	90	0000	0: SerDataH[0] 1: SerDataH[1] 2: SerDataH[2] 3: SerDataH[3]		Serial interface high data nibble
RegCCntl1	5B	91	0000	0: CountFSel0 1: CountFSel1 2: CountFSel2 3: Up/Down		10-bit counter control 1; frequency and up/down

Register Name	Add Hex	Add Dec.	Reset Value	Read Bits	Write Bits	Remarks
			b'3210	Read / Write Bits		
RegCCntl2	5C	92	0000	0: '0' 1: EnComp 2: EvCount 3: Start	0: Load 1: EnComp 2: EvCount 3: Start	10-bit counter control 2; comparison, event counter and start
RegCDataL	5D	93	0000	0: Count[0] 1: Count[1] 2: Count[2] 3: Count[3]	0: CReg[0] 1: CReg[1] 2: CReg[2] 3: CReg[3]	10-bit counter data low nibble
RegCDataM	5E	94	0000	0: Count[4] 1: Count[5] 2: Count[6] 3: Count[7]	0: CReg[4] 1: CReg[5] 2: CReg[6] 3: CReg[7]	10-bit counter data middle nibble
RegCDataH	5F	95	0000	0: Count[8] 1: Count[9] 2: BitSel[0] 3: BitSel[1]	0: CReg[8] 1: CReg[9] 2: BitSel[0] 3: BitSel[1]	10-bit counter data high bits
RegPA1	60	96	p000	0: PA[4] 1: PA[5] 2: OEnPA[5] 3: NchOpDrPA[5]	0: -- 1: PAout[5] 2: OEnPA[5] 3: NchOpDrPA[5]	PortA [5:4] Direct input read, Output data register with Output enable active Hi
RegPaCntl3	61	97	pppp	0: PWMoutPA[0] 1: PWMoutPA[1] 2: SerialCkPA [1] 3: SerialStPA [3]		PortA Control3 Output distribution on PA[0], PA[1] and PA[3]
RegPaCntl4	62	98	ppPP	0: freqOutPA[2] 1: Sout/rstPA[2] 2: : freqOutPA[5] 3: NoPdPA[5]		PortA Control4 Output distribution on PA[2] and PA[5]
RegIRQMask1	65	101	0000	0: MaskIRQPA[0/5] 1: MaskIRQPA[3/4] 2: MaskIRQCntComp 3: MaskIRQCount0		Port A & Counter interrupt mask; masking active 0
RegIRQMask2	66	102	0000	0: MaskIRQSerial 1: MaskIRQSvld 2: MaskIRQHz64/8 3: MaskIRQHz1		Prescaler, SVLD & serial interf. interrupt mask; masking active low
RegIRQ1	67	103	0000	0: IRQPA[0/5] 1: IRQPA[3/4] 2: IRQCntComp 3: IRQCount0	0:RIRQPA[0/5] 1:RIRQPA[3/4] 2:RIRQCntComp 3:RIRQCount0	Read: port A & Counter interrupt Write: Reset IRQ if data bit = 1.
REgIRQ2	68	104	0000	0: IRQSerial 1: IRQSvld 2: IRQHz64/8 3: IRQHz1	0:RIRQSerial 1:RIRQSvld 2:RIRQHz64/8 3:RIRQHz1	Read: Prescaler, SVLD & serial interface interrupt. Write: Reset IRQ if data bit = 1
RegSysCntl1	69	105	000p p = POR	*0: ChTmDis* 1: SelIntFull 2: '0' 3: IntEn	*0: ChTmDis* 1: SelIntFull 2: Sleep 3: IntEn	System control 1; *ChTmDis only usable only for EM test modes*
RegSysCntl2	6A	106	Pp00 p = POR	0: WDVal0 1: WDVal1 2: SleepEn 3: PORstatus	0: -- 1: -- 2: SleepEn 3: WDReset	System control 2; watchdog value and periodical reset, enable sleep mode
RegSleepCR	6B	107	pppp p = POR	0: SCRsel0 1: SCRsel1 2: SleepCntDis 3: NoPullPA[4]		Sleep Counter reset control

Register Name	Add Hex	Add Dec.	Reset Value	Read Bits	Write Bits	Remarks
			b'3210	Read / Write Bits		
RegPresc	6C	108	0000	0: DebSel 1: PrIntSel 2: '0' 3: ExtCPUclkON	0: DebSel 1: PrIntSel 2: ResPresc 3: ExtCPUclkON	Prescaler control; Debouncer, prescaler interrupt select and reset, External CPU clock enable
IXLow	6E	110	xxxx	0: IXLow[0] 1: IXLow[1] 2: IXLow[2] 3: IXLow[3]		Internal µP index register low nibble; for µP indexed addressing
IXHigh	6F	111	xxxx	0: IXHigh[4] 1: IXHigh[5] 2: IXHigh[6] 3: '0'	0: IXHigh[4] 1: IXHigh[5] 2: IXHigh[6] 3: --	Internal µP index register high nibble; for µP indexed addressing
RegVldCntl	73	115	000p	0: NoWDtim 1: SVLDen 2: VldBusy 3: VldResult	0: NoWDtim 1: SVLDen 2: VldStart 3: ExtVcheck	Voltage level detector & RC osc. control
RegSVLDlev	74	116	PpPp	0: SVLDlev[0] 1: SVLDlev[1] 2: SVLDlev[2] 3: SVLDlev[3]		SVLD test voltage level select
RegOscTrimH	75	117	pPPP	0: RegOsctrim[4] 1: RegOsctrim[5] 2: RegOsctrim[6] 3: RegOsctrim[7]		Oscillator trimming word, MSB
RegOscTrimL	76	118	PPPP	0: RegOsctrim[0] 1: RegOsctrim[1] 2: RegOsctrim[2] 3: RegOsctrim[3]		Oscillator trimming word, LSB
RegMFP0	79	121	pppp	0:Opt[0] 1:Opt[1] 2:Opt[2] 3:Opt[3]		PA4 pull up/down selection Counter clock option Serial Interface clock Debouncer clock
RegMFP1	7A	122	pppp	0:Opt[4] 1:Opt[5] 2:Opt[7] 3:Opt[7]		RC frequency base selection RC frequency selection RC frequency selection RC frequency selection
RegMFP2	7B	123	pppp	0:Opt[8] 1: 2: 3:		ADC/SVLD level #15 selection
	7C	124	xxxx			Reserved for EM6680 software compatibility.
	7D	125	xxxx			Reserved for EM6680 software compatibility.
RegTestEM1	7E	126	pppp	0: Tmsel[0] 1: Tmsel[1] 2: Tmsel[2] 3: disablePOR		For EM test only
RegTestEM2	7F	127	pppp	0: OeTm0 1: OeTm1 2: TestResSys 3: TestPOR		For EM test only

p = defined by POR at '0' (power on reset) only
P = defined by POR at '1' (power on reset) only
x = undefined state by reset (register must be written before used)
RegTestEm1 and RegTestEm2 can be written only if *ChTmDis* in RegSysCntl1 is '0'. They are used for EM test only and are Write only. To prevent entering test mode in normal operation one has to set the *ChTmDis* bit to '1' as soon as possible after reset. It should be one of first instructions.

14. Mask Options

Mask option are available on the EM6680, on EM6580 options are accessible through specific registers that are only present into the EM6580. These registers are name RegMFPn (n 0,1,2 and 3) and are mapped within a free range addresses, from address 75h to 7Bh

14.1 Port A Metal Options

Table 14.1.1 Pull-down Metal mask Options

	Description	**Strong Pull-down**	**Weak Pull-down**	**R1 Value Typ.100k**	**NO Pull-Down**
Option Name		1	2	3	4
MAPD[5]	PA[5] input pull-down	**X**		100k	
MAPD[4]	PA[4] input pull-down	**X**		100k	
MAPD[3]	PA[3] input pull-down	**X**		100k	
MAPD[2]	PA[2] input pull-down	**X**		100k	
MAPD[1]	PA[1] input pull-down	**X**		100k	
MAPD[0]	PA[0] input pull-down	**X**		100k	

NO pull down option is already software controllable with register as in the EM6680.

Table 14.1.2 Pull-up Metal mask Options

	Description	**Strong Pull-up**	**Weak Pull-up**	**R1 Value Typ.100k**	**NO Pull-up**
Option Name		1	2	3	4
MAPU[5]	PA[5] input pull-up	**X**		100k	
MAPU[4]	PA[4] input pull-up	**X**		100k	
MAPU[3]	PA[3] input pull-up	**X**		100k	
MAPU[2]	PA[2] input pull-up	**X**		100k	
MAPU[1]	PA[1] input pull-up	**X**		100k	
MAPU[0]	PA[0] input pull-up	**X**		100k	

NO pull up option is already software controllable with register as in the EM6680.

PA[4] pull -up pull-down selection can be down with **regMFP0 Opt[0].**

14.2 RC oscillator Frequency Option

Option Name	**Description**	**Default Value**	**User Value**
		A	**B**
RCfreq	RC osc Frequency	**32 kHz**	

By default the RC oscillator frequency is typ. 32 kHz With option RCfreq. Other possibilities are: 64kHz, 128kHz, 256kHz and 512kHz or 50kHz, 100kHz, 200kHz, 400kHz or max. 800kHz .

RC frequency selection is done with register **regMFP1 Opt[7:4]**

14.3 Debouncer Frequency Option

Option Name	Description	Default Value	User Value
		A	B
MDeb	Debouncer freq.	**Ck[11]**	

By default the debouncer frequency is Ck[11]. The user may choose Ck[14] instead of Ck[11].
Ck[14]corresponds to maximum 0.25ms debouncer time in case of a 32kHz System Clock – SysClk.

Debouncer Frequency is selected with register **RegMFP0 Opt[3**].

14.4 Power-Check Level Option

Option Name	Description	Default Value	User Value
		A	B
PClev.	Power-Check level	**2.00**	**NA**

The power check level is fixed at 2V (Typ) v in order to guarantee a proper working of the Flash memory.

14.5 ADC/SVLD Voltage Level #15

Option Name	Description	Default Value	User Value
		A	B
HisvldLev.	ADC/SVLD lev.#15	**2.75V**	

By default the ADC/SVLD Voltage Level #15 is at 2.75V but user can select also the second possibility which is 3.0V and does not influence on other ADC/SVLD levels.

Selection is done with register **RegMFP2 Opt[0]**.

14.6 Counter Update option

Option Name		Default Value	User Value
		A	B
CntF	Counter Reg. level	**SysClk**	

By default the counter is updated by Sysclk (32 or 50kHz typ) and the highest counter frequency is Sysclk/2 (16 or 25kHz Typ). The other possibility is to select RCclk for counter update freq. Which gives a possibility to replace port A input at selection 7 by RC/2. If **DebouncerNoPA[3/4]** in **RegPACntl[1]** is '1' then the resulting clock on timer selection 7 is CPUClk divided by 4. If **DebouncerNoPA[3/4]** is '0' then the resulting clock is CpuClk/2 directly.

Counter Update Option selection is done with register **RegMFP0 Opt[1]**

14.7 Voltage Regulator level option

Name		Default Value	User Value
		A	B
LevelReg	Voltage Reg. level	**2T/W**	**NA**

This is not an option any more.

14.8 Additional registers compare to EM6680

This paragraph describe new registers with respect to EM6680. These registers are the substitute of metal options we can find on the EM6680.

Register **RegOscTrimH** @ addr. 117 decimal = 75 hex.

Bit	Name	Reset	R/W	Description
3	RegOscTrim[7]	0	R/W	Oscillator trimming register, MSB
2	RegOscTrim[6]	1	R/W	
1	RegOscTrim[5]	1	R/W	
0	RegOscTrim[4]	1	R/W	

Register **RegOscTrimL** @ addr. 118 decimal = 76 hex.

Bit	Name	Reset	R/W	Description
3	RegOscTrim[3]	1	R/W	Oscillator trimming register, LSB
2	RegOscTrim[2]	1	R/W	
1	RegOscTrim[1]	1	R/W	
0	RegOscTrim[0]	1	R/W	

Register **RegMFP0** @ addr. 121 decimal = 79 hex.

Bit	Name	Reset	R/W	Description
3	Opt[3]	0	R/W	Debouncer clock select “0” = ck[11], “1”=ck[14]
2	Opt[2]	0	R/W	Must be remain to “0”
1	Opt[1]	0	R/W	Counter Clock source 7 selection "0" PA3/PA4, "1" RCclk/2
0	Opt[0]	0	R/W	PA4 Pull up/Pull down selection, "0" P-down, "1" P-up

Note1. bit1: When Opt[1]=1 the clock of the counter is no more ck16 (32kHz/50kHZ) but the CPU clock in order to achieve a correct sampling of the RCclk/2 signal.

Register **RegMFP1** @ addr. 122 decimal = 7A hex.

Bit	Name	Reset	R/W	Description
3	Opt[7]	0	R/W	RC base frequency (32/50 kHz) selection.
2	Opt[6]	0	R/W	RC frequency selection bit 2
1	Opt[5]	0	R/W	RC frequency selection bit 1
0	Opt[4]	0	R/W	RC frequency selection bit 0

Register **RegMFP2** @ addr. 123 decimal = 7B hex.

Bit	Name	Reset	R/W	Description
3	Opt[11]	?	R/W	
2	Opt[10]	?	R/W	
1	Opt[9]	?	R/W	
0	Opt[8]	0	R/W	ADC/SVLD voltage level#15: "0" 2.75v, "1" 3v

15. RC Oscillator

The oscillator is able to generate 2 main frequencies, 800kHz and 512kHz, then through a division chain generates the 12 possible frequencies of EM6680 chip. The frequency selection as opposed to EM6680 in done with registers, **RegMFP1**.

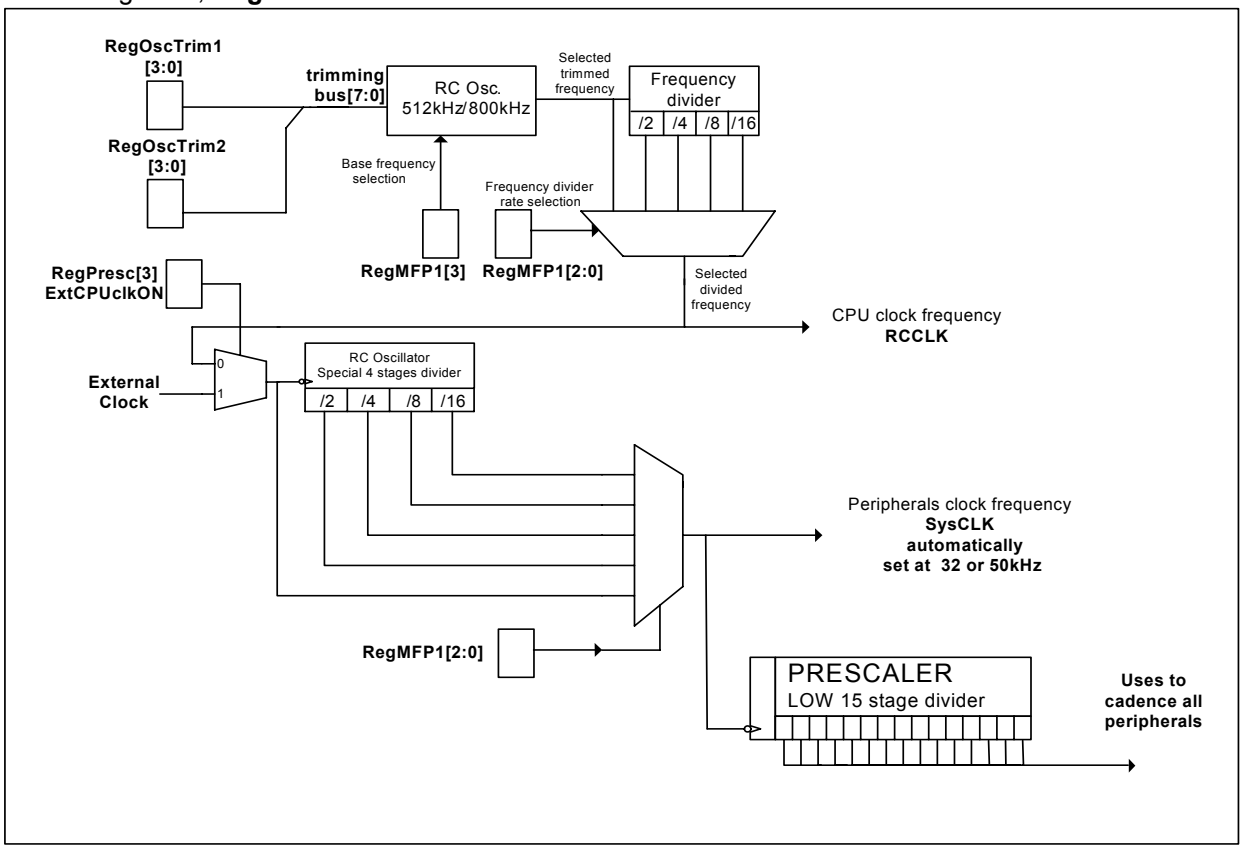


Figure 26: Oscillator architecture

15.1 Frequency selection

After power up the oscillator runs at 32 kHz, frequency switching is done by the software thanks to a register. This frequency switching has to be done in a way it guarantees a proper working of the microcontroller.

RC Oscillator frequency selection. **RegMFP1**[3:0]

RegMFP1[3]	RegMFP1[2]	RegMFP1[1]	RegMFP1[0]	Frequency of RCclk	Unit
Opt[7]	Opt[6]	Opt[5]	Opt[4]		
0	0	0	0	**32.00**	**kHz**
0	0	0	1	**64.00**	**kHz**
0	0	1	0	**128.00**	**kHz**
0	0	1	1	**256.00**	**kHz**
0	1	0	0	**512.00**	**kHz**
1	0	0	0	**50.00**	**kHz**
1	0	0	1	**100.00**	**kHz**
1	0	1	0	**200.00**	**kHz**
1	0	1	1	**400.00**	**kHz**
1	1	0	0	**800.00**	**kHz**

RegMFP1[3] bit selects the base frequency, 512kHz or 800kHz.
RegMFP1[2:0] bit select the frequency divider rate, 1, 2, 4, 8, 16.

When the RC frequency is changed the system clock, sysClk, has to be as close as possible to 32/50 kHz thanks to frequency dividers.

15.2 Oscillator Trimming

The internal oscillator circuit is designed for use with no external components to provide a clock source with tolerance less than ±25%. Two 4-bit registers, **RegOscTrimH** and **RegOscTrimL** allow adjustment to a tolerance less than ±5%. The default value of the trimming register is 7Fh allowing a frequency adjustment from –127 steps to +128 steps around this central value. Increasing the trimming register increases the frequency of the oscillator.

Reserved Flash location, **32** words at the end of the flash memory, contains the value of the trimming and the function allowing to load the register with the trimming value since the 4-bit microcontroller core does not have indirect addressing to code space.

EM6580 Flash Memory is splitted in two areas. First area (Sector 1) contains the main application code whereas the second are (Sector 2) contains specific trimming code for the RC oscillator. Manufactory trimming code for 512kHz and 800kHz frequencies are defined as follow :

Address	Memory area
0000H 0FDFH	Sector 1 Application code area
0FE0H . . . 0FFFH	Sector 2 Trimming code area

```
Address                                    Op-codes
        ; 512kHz Trimming function
0FE0H   STI     RegOscTrimH, 0xH           Cx75
0FE1H   STI     RegOscTrimL, 0xH           Cx76
0FE2H   RET                                FA7F
        ; 800kHz Trimming function
0FE3H   STI     RegOscTrimH, 0xH           Cx75
0FE4H   STI     RegOscTrimL, 0xH           Cx76
0FE5H   RET                                FA7F
```

x represents the trimming value
(differ from chip to chip)

To be able to use this routine, you need to define two constants which define the Trimming code location for each trim routine.

```
Trim_osc_base_512kHz    EQU  4064    (0FE0H)
Trim_osc_base_800kHz    EQU  4067    (0FE3H)
```

If you need to use trimming for your the application, just use a call instruction to trim the RC oscillator at the correct frequency.

```
CALL            Trim_osc_base_512kHz
        or
CALL            Trim_osc_base_800kHz
```

16. Unique ID Code / serial number

A unique ID code is included into the sector 2 of the flash memory during the manufacturing tests. Thanks to this unique code, the EM6580 is suitable for applications such as for instance logistics and tracking.

This code is made of 52 bits unique code and 16 bits CRC for validation, see definition underneath:

Address	Memory area
0000H 0FDFH	Sector 1 Application code area
0FE0H 0FFFH	Sector 2 area

Address			*Op-codes*
0FE6H	STI	3FH,0***x***H ; CRC4 (checksum)	*C**x**3F*
0FE7H	STI	40H,0***x***H ; CRC3 (checksum)	*C**x**40*
0FE8H	STI	41H,0***x***H ; CRC2 (checksum)	*C**x**41*
0FE9H	STI	42H,0***x***H ; CRC1 (checksum)	*C**x**42*
0FEAH	STI	43H,0***x***H ; KHH (tester N° High nibble)	*C**x**43*
0FEBH	STI	44H,0***x***H ; KLH (tester N° Low nibble)	*C**x**44*
0FECH	STI	45H,0***x***H ; MH (Month of the test)	*C**x**45*
0FEDH	STI	46H,0***x***H ; DHH (Day of test High nibble)	*C**x**46*
0FEEH	STI	47H,0***x***H ; DLH (Day of test Low nibble)	*C**x**47*
0FEFH	STI	48H,0***x***H ; YHH (Year of test High nibble)	*C**x**48*
0FF0H	STI	49H,0***x***H ; YLH (Year of test Low nibble)	*C**x**49*
0FF1H	STI	4AH,0***x***H ; HHH (Hour of test High nibble)	*C**x**4A*
0FF2H	STI	4BH,0***x***H ; HLH (Hour of test nibble)	*C**x**4B*
0FF3H	STI	4CH,0***x***H ; MHH (Minute of test High	*C**x**4C*
0FF4H	nibble)		*C**x**4D*
0FF5H	STI	4DH,0***x***H ; MLH (Minute of test Low nibble)	*C**x**4E*
0FF6H	STI	4EH,0***x***H ;SHH (Second of test High nibble)	*C**x**4F*
0FF7H	STI	4FH,0***x***H ; SLH (Second of test Low	*FA7F*
	nibble)		
	RET		

x represents the ID code value (unique for each chip)

To be able to use this routine, one needs to define a constant which define the date code location to restore the ID code into the ram address 3FH to 4FH.(The polynomial for the checksum can be provide to customer on request.)

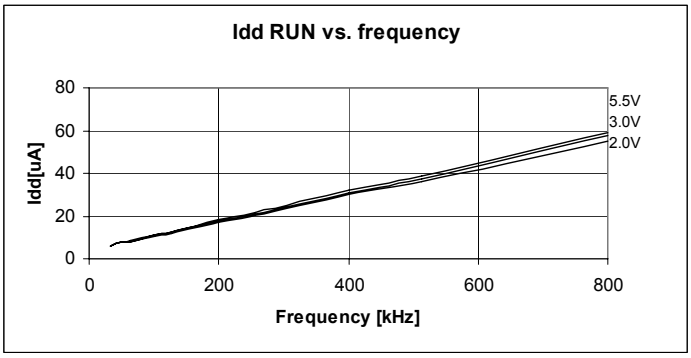
```
Date_Code          EQU   4070   (0FE6H)
```

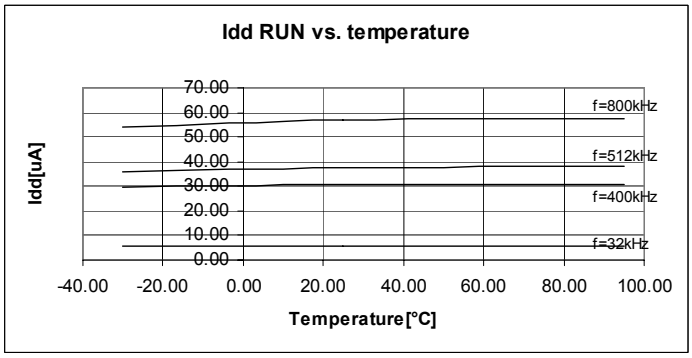
If you need to use the ID unique code in your application, just use a call instruction to the date_code label as follow :

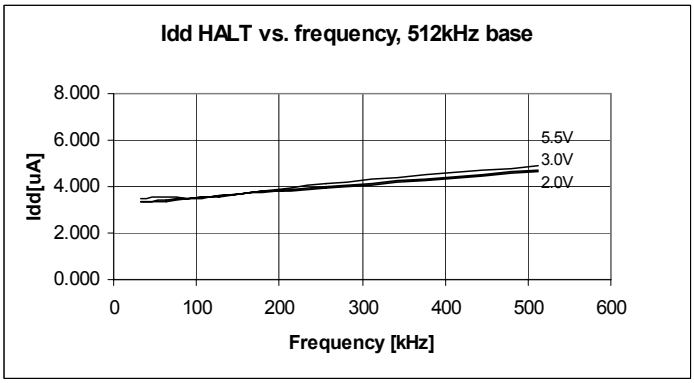
```
CALL   Date_Code
```

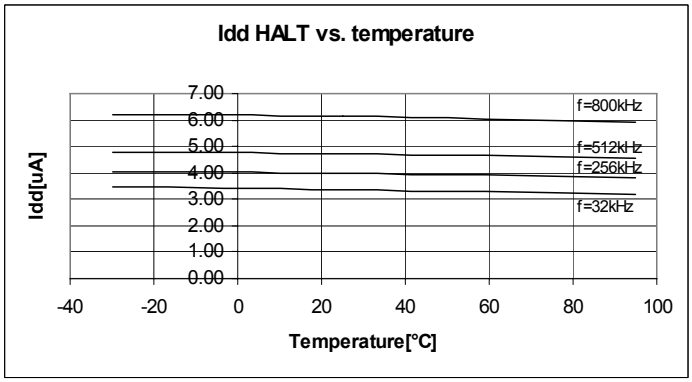
17. Temperature and Voltage Behaviour

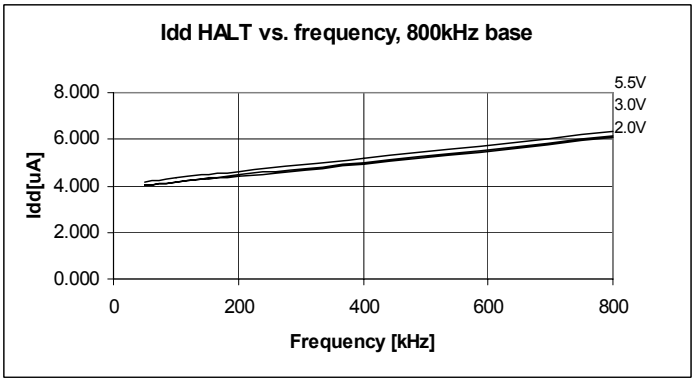
17.1 IDD current (Typical)

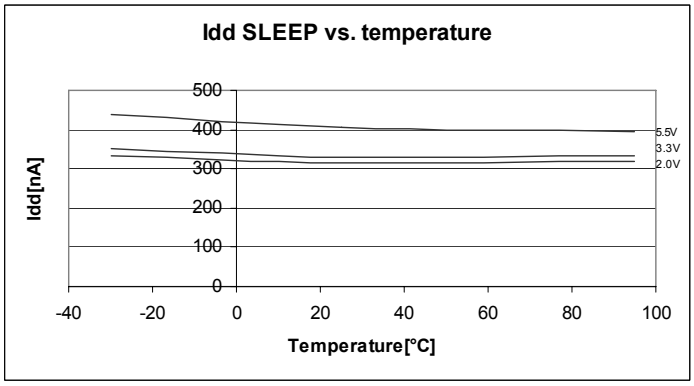




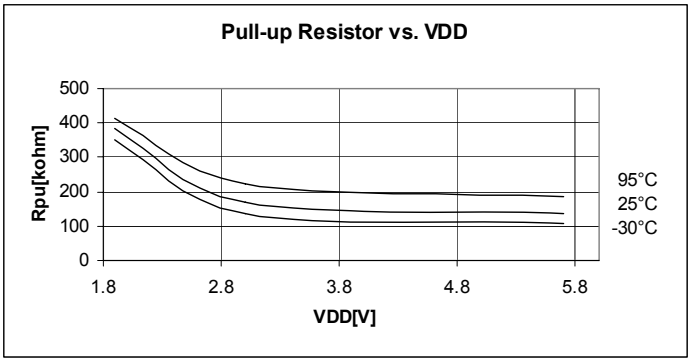


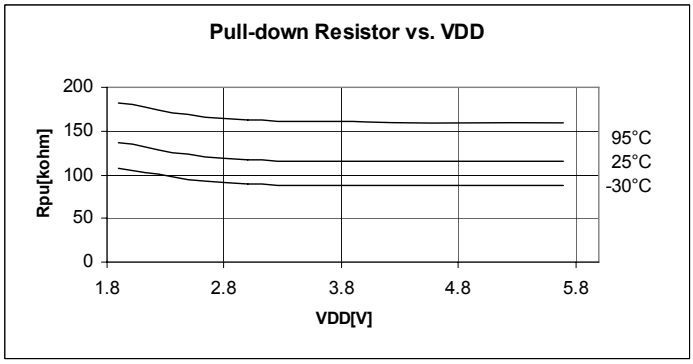




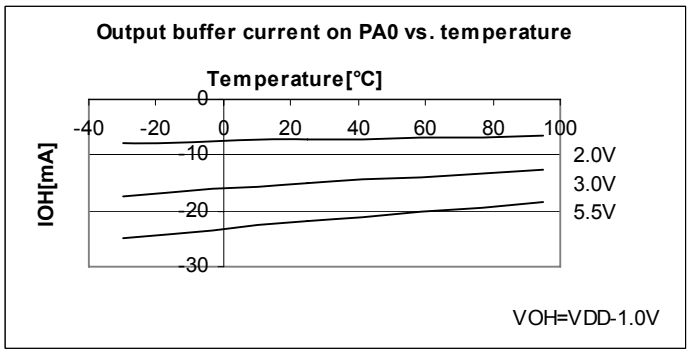


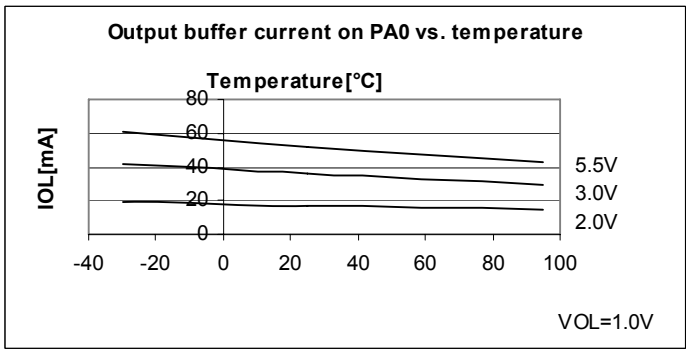
17.2 Pull-up and Pull-down resistors (Typical)

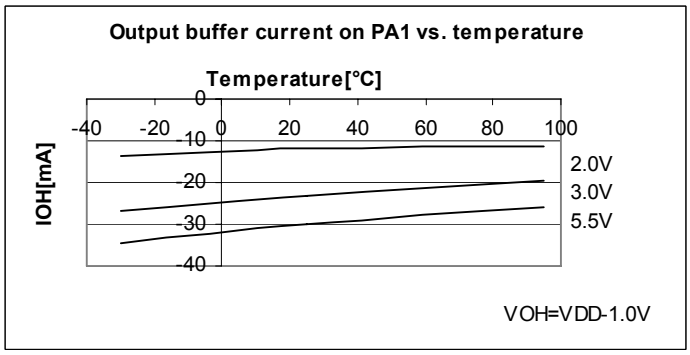


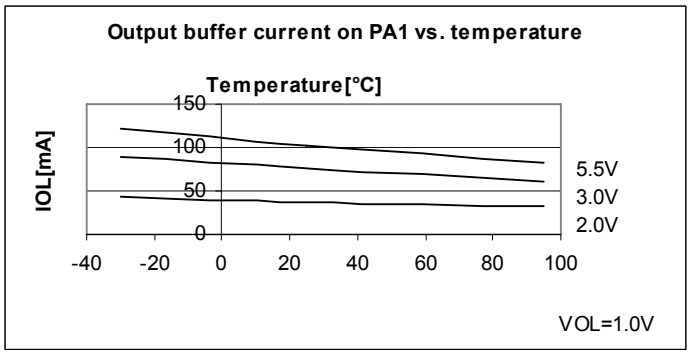


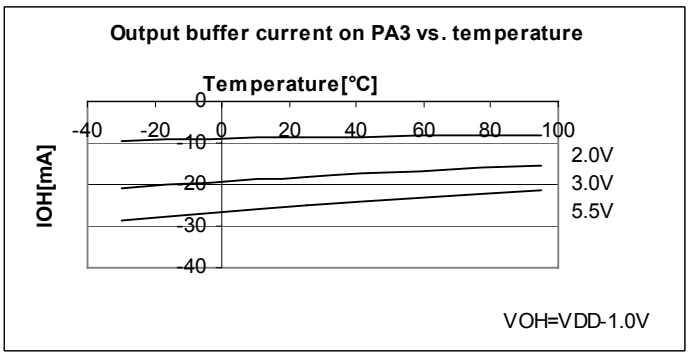
17.3 Output current (Typical)

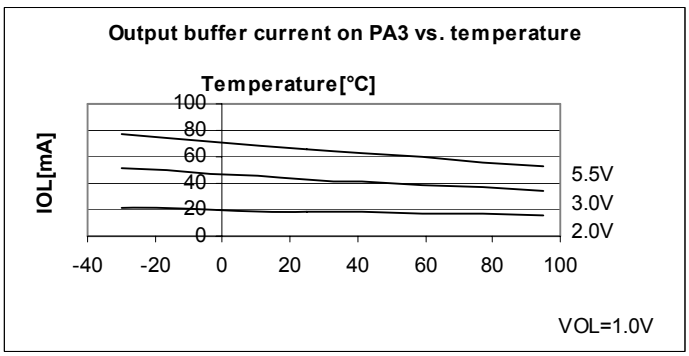




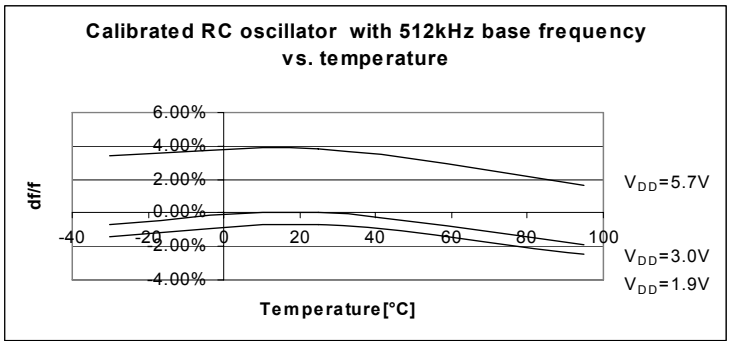


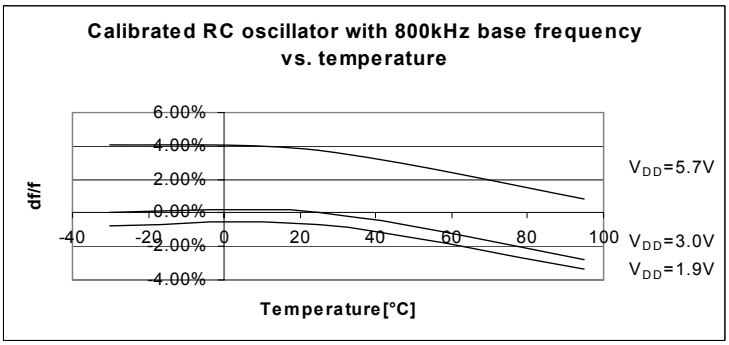






17.4 Oscillator Frequency (Typical)





18. Electrical Specification

18.1 Absolute Maximum Ratings

	Min.	Max.	Units
Power supply V_{DD}-V_{SS}	- 0.2	+ 6.0	V
Input voltage	V_{SS} – 0.2	V_{DD}+0.2	V
Storage temperature	- 40	+ 125	°C
Electrostatic discharge to Mil-Std-883C method 3015.7 with ref. to V_{SS}	-2000	+2000	V
Maximum soldering conditions Packages are Green-Mold and Lead-free	As per Jedec J-STD-020C		

Stresses above these listed maximum ratings may cause permanent damage to the device.
Exposure beyond specified electrical characteristics may affect device reliability or cause malfunction.

18.2 Handling Procedures

This device has built-in protection against high static voltages or electric fields; however, anti-static precautions should be taken as for any other CMOS component.
Unless otherwise specified, proper operation can only occur when all terminal voltages are kept within the supply voltage range.

18.3 Standard Operating Conditions

Parameter	MIN	TYP	MAX	Unit	Description
Temperature	-20	25	85	°C	
V_{DD} _Range1	2.0	3.0	5.5	V	Application working range
V_{DD} _Range2	2.7	3.0	3.3	V	Flash programming range
Vreg		1.9		V	Regulated Voltage
VPP		15		V	High Voltage for Flash programming
Cext	470			nF	External capacitor on Vreg port

18.4 DC Characteristics - Power Supply

Conditions: T=25°C, Frequency 32kHz

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
POR static level	-20 to 85°C	VPOR		1.50	1.80	V
Power-Check level	-20 to 85°C	VPC	1.68	2.00	2.30	V
RAM data retention		Vrd1	1.0			V

Conditions: V_{dd} =3.0V, Frequency 32kHz

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
ACTIVE Supply Current	25°C	I_{VDDa2}		5.8	7.0	µA
	-20 to 85°C	I_{VDDa2}			8.0	µA
STANDBY Supply Current	25°C	I_{VDDh2}		3.3	4.5	µA
	-20 to 85°C	I_{VDDh2}			5.5	µA
SLEEP Supply Current	25°C	I_{VDDs1}		0.32	0.6	µA
	-20 to 85°C	I_{VDDs2}			1.5	µA

Conditions: V_{DD}=3.0V, Frequency = 800kHz

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
ACTIVE Supply Current	25°C	I_{VDDa2}		56	75	µA
	-20 to 85°C	I_{VDDa2}			100	µA
STANDBY Supply Current	25°C	I_{VDDh2}		6.1	7.5	µA
	-20 to 85°C	I_{VDDh2}			8.5	µA
SLEEP Supply Current	25°C	I_{VDDs1}		0.32	0.6	µA
	-20 to 85°C	I_{VDDs2}			1.5	µA

18.5 Supply Voltage Level Detector

Conditions: V_{dd} =3.0V, T=25°C, rising signal on PA[4] and V_{DD}.

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
ADC voltage Level0	-20 to 85°C	V_{VLD0}	0.420	0.500	0.580	V
ADC voltage Level1	-20 to 85°C	V_{VLD1}	0.550	0.650	0.750	V
ADC voltage Level2	-20 to 85°C	V_{VLD2}	0.680	0.80	0.900	V
ADC voltage Level3	-20 to 85°C	V_{VLD3}	0.800	0.95	1.100	V
ADC voltage Level4	-20 to 85°C	V_{VLD4}	0.930	1.10	1.260	V
ADC voltage Level5	-20 to 85°C	V_{VLD5}	1.050	1.25	1.450	V
ADC voltage Level6	-20 to 85°C	V_{VLD6}	1.180	1.40	1.600	V
ADC voltage Level7	-20 to 85°C	V_{VLD7}	1.300	1.55	1.780	V
ADC voltage Level8	-20 to 85°C	V_{VLD8}	1.430	1.70	1.950	V
ADC voltage Level9	-20 to 85°C	V_{VLD9}	1.550	1.85	2.120	V
ADC/SVLD voltage Level10	-20 to 85°C	V_{VLD10}	1.680	2.00	2.300	V
ADC/SVLD voltage Level11	-20 to 85°C	V_{VLD11}	1.800	2.15	2.460	V
ADC/SVLD voltage Level12	-20 to 85°C	V_{VLD12}	1.950	2.30	2.640	V
ADC/SVLD voltage Level13	-20 to 85°C	V_{VLD13}	2.080	2.45	2.800	V
ADC/SVLD voltage Level14	-20 to 85°C	V_{VLD14}	2.200	2.60	2.980	V
ADC/SVLD voltage Level15	-20 to 85°C	V_{VLD15}	2.340	2.75	3.150	V
ADC/SVLD voltage Level15b	-20 to 85°C	V_{VLD15b}	2.550	3.00	3.420	V
ADC/SVLD negative hysteresis		V_{vld_hsyt}		50		mV
ADC input impedance		R_{adc_pa4}		1		MΩ

18.6 DC characteristics - I/O Pins

Conditions: T= -20 to 85°C (unless otherwise specified) V_{DD} =3.0V

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
Input Low voltage						
Port A[5:0]		V_{IL}	V_{ss}		0.2 V_{DD}	V
Input High voltage						
Port A[5:0]		V_{IH}	0.7 V_{DD}		V_{DD}	V
Input Pull-down						
PA[5:0] strong	V_{DD} =3.0V, Pin at 3.0V, 25°C	R_{PD}	80	125	180	kΩ
Input Pull-up						
PA[5:0] strong	V_{DD} =3.0V, Pin at 0.0V, 25°C	R_{PU}	80	175	280	kΩ
Output Low Current PA[0]	V_{DD} =3.0V , V_{OL} =0.15V	I_{OL}		6.7		mA
	V_{DD} =3.0V , V_{OL} =0.30V	I_{OL}		13.1		mA
	V_{DD} =3.0V , V_{OL} =0.50V	I_{OL}		20.8		mA
	V_{DD} =3.0V , V_{OL} =1.00V	I_{OL}	20.0	36.0		mA
Output Low Current PA[2,1]	V_{DD} =3.0V , V_{OL} =0.15V	I_{OL}		13.6		mA
	V_{DD} =3.0V , V_{OL} =0.30V	I_{OL}		26.4		mA
	V_{DD} =3.0V , V_{OL} =0.50V	I_{OL}		42.6		mA
	V_{DD} =3.0V , V_{OL} =1.00V	I_{OL}	45.0	73.0		mA
Output Low Current PA[3]	V_{DD} =3.0V , V_{OL} =0.15V	I_{OL}		8.2		mA
	V_{DD} =3.0V , V_{OL} =0.30V	I_{OL}		15.9		mA
	V_{DD} =3.0V , V_{OL} =0.50V	I_{OL}		25.0		mA
	V_{DD} =3.0V , V_{OL} =1.00V	I_{OL}	25.0	42.7		mA
Output Low Current PA[5]	V_{DD} =3.0V , V_{OL} =0.15V	I_{OL}		7.6		mA
	V_{DD} =3.0V , V_{OL} =0.30V	I_{OL}		14.8		mA
	V_{DD} =3.0V , V_{OL} =0.50V	I_{OL}		23.5		mA
	V_{DD} =3.0V , V_{OL} =1.00V	I_{OL}	20.0	39.0		mA

Conditions: T= -20 to 85°C (unless otherwise specified) V_{DD} =3.0V

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
Output High Current PA[0]	V_{DD} =3.0V, V_{OH} = V_{DD} -0.15V	I_{OH}		-2.75		mA
	V_{DD} =3.0V , V_{OH} = V_{DD} -0.30V	I_{OH}		-5.3		mA
	V_{DD} =3.0V , V_{OH} = V_{DD} -0.50V	I_{OH}		-8.5		mA
	V_{DD} =3.0V , V_{OH} = V_{DD} -1.00V	I_{OH}		-15.3	-10.0	mA
Output High Current PA[2,1]	V_{DD} =3.0V, V_{OH} = V_{DD} -0.15V	I_{OH}		-4.0		mA
	V_{DD} =3.0V , V_{OH} = V_{DD} -0.30V	I_{OH}		-7.7		mA
	V_{DD} =3.0V , V_{OH} = V_{DD} -0.50V	I_{OH}		-12.7		mA
	V_{DD} =3.0V , V_{OH} = V_{DD} -1.00V	I_{OH}		-23.6	-15.0	mA
Output High Current PA[3]	V_{DD} =3.0V, V_{OH} = V_{DD} -0.15V	I_{OH}		-3.2		mA
	V_{DD} =3.0V , V_{OH} = V_{DD} -0.30V	I_{OH}		-6.3		mA
	V_{DD} =3.0V , V_{OH} = V_{DD} -0.50V	I_{OH}		-10.1		mA
	V_{DD} =3.0V , V_{OH} = V_{DD} -1.00V	I_{OH}		-18.5	-12.0	mA
Output High Current PA[5]	V_{DD} =3.0V, V_{OH} = V_{DD} -0.15V	I_{OH}		-3.2		mA
	V_{DD} =3.0V , V_{OH} = V_{DD} -0.30V	I_{OH}		-6.0		mA
	V_{DD} =3.0V , V_{OH} = V_{DD} -0.50V	I_{OH}		-9.5		mA
	V_{DD} =3.0V , V_{OH} = V_{DD} -1.00V	I_{OH}		-17.2	-11.0	mA

18.7 RC oscillator frequency

Conditions: V_{DD} =3.0V, T=25°C (unless otherwise specified)

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
512 kHz untrimmed		f512	-40%	512	+40%	kHz
512 kHz trimmed		f512tr	-5%	512	+5%	kHz
	-20 to 85°C	f512tr	-10%	512	+10%	kHz
800 kHz untrimmed		f800	-40%	800	-40%	kHz
800 kHz trimmed		f800tr	-5%	800	+5%	kHz
	-20 to 85°C	f800tr	-10%	800	+10%	kHz
Oscillator start voltage		Ustart	V_{DDmin}			V
System start time (oscillator + cold-start + reset)	V_{DD} > V_{DDMin}	tdsys		0.5	6.0	ms

18.8 Sleep Counter Reset - SCR

Conditions: V_{DD} =3.0V

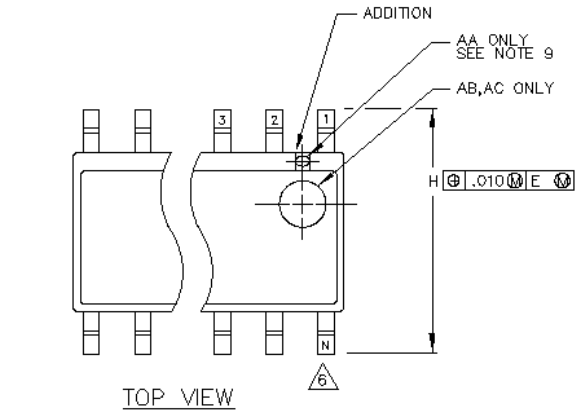
Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
SCR timeout 0	-20 to 85°C	tSCR00	12	23	34	ms
SCR timeout 1	-20 to 85°C	tSCR01	123	237	350	ms
SCR timeout 2	-20 to 85°C	tSCR10	1.01	1.95	2.88	s
SCR timeout 3	-20 to 85°C	tSCR11	8.2	15.7	23.21	s

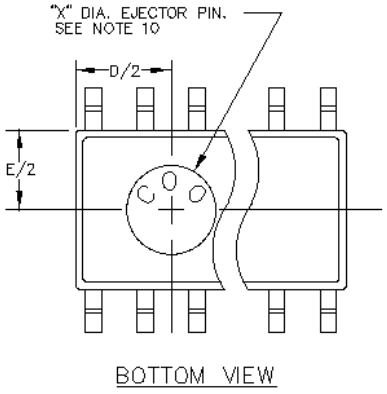
19. Pad Location Diagram

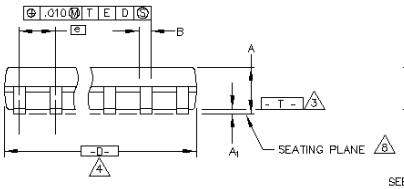
Information upon request to EM Microelectronic-Marin SA.

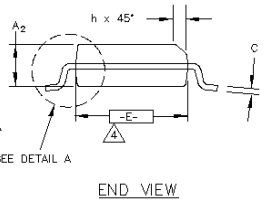
20. Package Dimensions

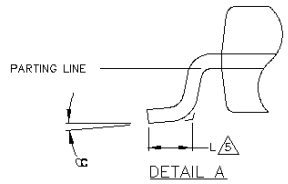
20.1 SO-8/14











NOTES:

1. MAXIMUM DIE THICKNESS ALLOWABLE IS .015.
2. DIMENSIONING & TOLERANCES PER ANSI.Y14.5M - 1982.
3. "T" IS A REFERENCE DATUM.
4. "D" & "E" ARE REFERENCE DATUMS AND DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS, BUT DOES INCLUDE MOLD MISMATCH AND ARE MEASURED AT THE MOLD PARTING LINE. MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED 0.006 INCHES AT END AND .010 INCHES AT WINDOW
5. "L" IS THE LENGTH OF TERMINAL FOR SOLDERING TO A SUBSTRATE.
6. "N" IS THE NUMBER OF TERMINAL POSITIONS.
7. TERMINAL POSITIONS ARE SHOWN FOR REFERENCE ONLY.
8. FORMED LEADS SHALL BE PLANAR WITH RESPECT TO ONE ANOTHER WITHIN .003 INCHES AT SEATING PLANE.
9. THE APPEARANCE OF PIN #1 I.D ON THE 8 LD IS OPTIONAL, ROUND TYPE ON SINGLE LEADFRAME AND RECTANGULAR TYPE ON MATRIX LEADFRAME.
10. COUNTRY OF ORIGIN LOCATION ON PACKAGE BOTTOM IS OPTIONAL AND DEPENDS ON ASSEMBLY LOCATION.
11. CONTROLLING DIMENSION: INCHES.
12. THIS PART IS COMPLIANT WITH JEDEC STANDARD MS-012, VARIATION AA, AB & AC.

THIS TABLE IN MILLIMETERS

SYMBOL	COMMON DIMENSIONS			NOTE	NOTE VARIATIONS	3 D			5 N
	MIN.	NOM.	MAX.			MIN.	NOM.	MAX.	
A	1.55	1.63	1.73		AA	4.80	4.93	4.98	8
A_1	0.127	0.15	0.25		AB	8.58	8.69	8.74	14
A_2	1.40	1.47	1.55		AC	9.80	9.93	9.98	16
B	0.35	0.41	0.49						
C	0.19	0.20	0.25						
D	SEE VARIATIONS			3					
E	3.81	3.94	3.99						
e	1.27 BSC								
H	5.84	5.99	6.20						
h	0.25	0.33	0.41						
L	0.41	0.64	0.89						
N	SEE VARIATIONS			5					
α	0°	5°	8°						
X	2.16	2.36	2.54						

21. Ordering Information

Packaged Device:

EM6580 SO8 A+

Package:
SO8 = 8 pin SOIC
SO14 = 14 pin SOIC

Delivery Form:
A = Stick
B = Tape&Reel

Device in DIE Form:

EM6580 WS 11

Die form:
WW = Wafer
WS = Sawn Wafer/Frame
WP = Waffle Pack

Thickness:
11 = 11 mils (280um), by default
27 = 27 mils (686um), not backlapped
(for other thickness, contact EM)

In its packaged form, EM6580 comes in green mold /lead free (symbolized by a "+" at the end of the part number).

Ordering Part Number (selected examples)

Part Number	Package/Die Form	Delivery Form/ Thickness
EM6580SO8A +	8 pin SOIC	Stick
EM6580SO8B +	8 pin SOIC	Tape&Reel
EM6580SO14A +	14 pin SOIC	Stick
EM6580SO14B +	14 pin SOIC	Tape&Reel
EM6580WS11	Sawn wafer	11 mils
EM6580WP11	Die in waffle pack	11 mils

Please make sure to give the complete Part Number when ordering.

21.1 Package Marking

8-pin SOIC marking:

First line:	6	5	8	0	0	1	
Second line:	P	P	P	P	P	P	
Third line:	R	1	A			Y	

14-pin SOIC marking:

First line:	E	M	6	5	8	0	0	1	
Second line:	P	P	P	P	P	P	P	P	
Third line:	R	1	A				Y	P	

Where: PP…P = Production identification (date & lot number) of EM Microelectronic-Marin SA
Y = year of assembly