Low Power Microcontroller with EEPROM AND RC Oscillator

Features

- Low Power - 42µA active mode
  - 8µA standby mode
  - 0.3µA sleep mode
  @ 3.0V, 600kHz, 25°C, typ
- Voltage Range - 1.9 to 5.5 V
- Supply voltage level detection (SVLD)
- ROM - 1280 × 16 bit
- RAM - 80 × 4 bit
- EEPROM - 32 x 8 bit
- 2 clocks per instruction cycle
- 72 basic instructions
- RC oscillator
- Oscillation detection circuit / Digital watchdog timer reset.
- Maximum 12 inputs (3 ports)
- Maximum 8 outputs (2 ports)
- Serial Write Buffer - 256 bits (SWB)
- 10 bit up/down counter with PWM capability
- Frequency out 600kHz, 37.5kHz, 2.3kHz, PWM
- Sleep Counter Reset (SCR) programmable.
- 8 internal interrupt sources (3×prescaler, 2×timer ,1xSWB, 1×SVLD, 1xEEPROM)
- 4 external interrupt sources (port A)
- Reset with input combinations
- Packages available : TSSOP16, SO16, SO18

Description

The EM6640 is an advanced single chip CMOS 4-bit microcontroller. It contains ROM, RAM, EEPROM, watchdog timer, oscillation detection circuit, 10 bit up/down counter, prescaler, supply voltage level detector (SVLD), sleep counter reset (SCR), frequency output and SWB.
The low voltage feature and low power consumption make it the most suitable controller for battery, stand alone and mobile equipment. The EM66XX series is manufactured using EM Microelectronic's Advanced Low Power (ALP) CMOS Process.

Typical Applications

- remote controls
- medical applications
- domestic appliance
- safety and security devices
- measurement equipment
- interactive system
- keyless entry with rolling code

Figure 1 Architecture

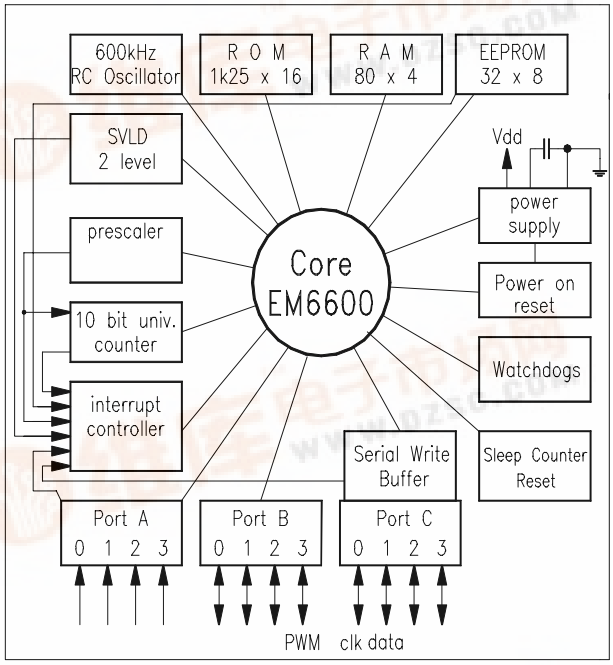
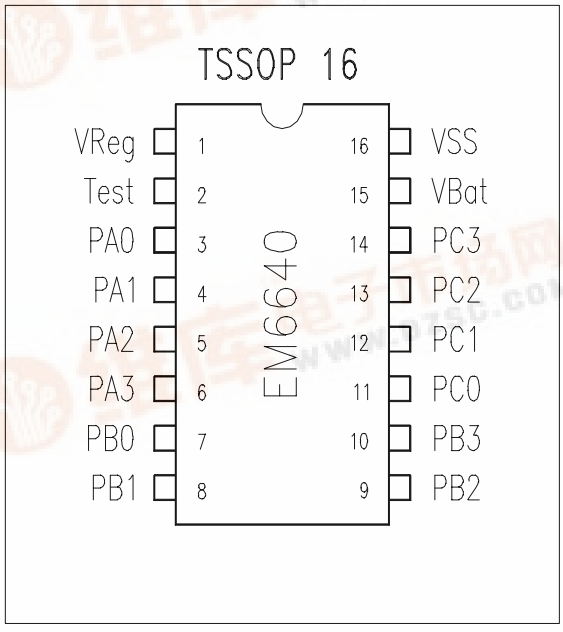


Figure 2 Pin Configuration of TSSOP16



EM6640 at a glance

- **Power Supply**
  - Low voltage, low power architecture including internal voltage regulator.
  - 1.9V ... 5.5V battery voltage.
  - 600 kHz RC oscillator.
  - 42µA typical in active mode @ 3V, 25C°.
  - 8µA typical in standby mode @ 3V, 25C°.
  - 0.3µA typical in sleep mode @ 3V, 25C°.

- **RAM**
  - 80 x 4 bit, directly addressable.

- **ROM**
  - 1280 x 16 bit metal mask programmable.

- **EEPROM**
  - 32 x 8 bit, indirectly addressable (6 bits used to adjust the oscillator frequency).
  - Interrupt request at the end of writing operation.
  - 60µA typical during read mode @ 3V, 25C°.
  - 45µA typical during erase/write mode @ 3V, 25C°

- **CPU**
  - 4 bits RISC architecture.
  - 2 clock cycles per instruction.
  - 72 basics instructions.

- **Main Operating Modes and Resets**
  - Active mode (CPU is running).
  - Standby mode (CPU in halt).
  - Sleep mode (No clock, reset state).
  - Initial reset on power on (POR).
  - Watchdog timer (time out) reset.
  - Oscillation detection watchdog reset.
  - Reset with input combination.

- **4 Bits Input Port A**
  - Direct input read.
  - Reset with input combination (register selectable).
  - Debounced or direct input (register selectable).
  - Interrupt request on input's rising or falling edge (register selectable).
  - Pull-up, pull-down or none (register selectable).
  - Software test variables for conditional jumps.
  - PA[0] and PA[3] are input for the event counter.

- **4 Bits Input/Output Port B**
  - Input or Output port bitwise.
  - Direct input read.
  - CMOS or N-channel open drain outputs.
  - Pull-up selectable in N-channel open drain mode.
  - Pull-down or pull-up selectable by register.
  - Selectable pulse width modulation (PWM).
  - PWM output on PB[3].
  - Output frequencies 600kHz, 37.5kHz, 2.3kHz.

- **4 Bits Input/Output Port C**
  - Input or output port bitwise.
  - Direct input read.
  - CMOS or N-channel open drain outputs.
  - Pull-up selectable in N-channel open drain mode.
  - Pull-down or pull-up selectable by register.
  - Serial Write Buffer clock and data output.

- **Oscillator**
  - RC Oscillator at f=600kHz ±1% typ (-30°C...40°C).
  - Absolute frequency adjustable with 6 bits EEPROM.
  - No external components are necessary.

- **Serial Write Buffer (SWB)**
  - Max. 256 bits long clocked with 150kHz; 75kHz 9.4kHz; 2,3kHz. External clock capability in automatic mode, max: 1.5MHz.
  - Automatic send mode: number of clocks of the last nibble selectable by register and last data level latched. External clock division capability by 1/1, 1/4, 1/88 and 1/352.
  - Interactive send mode: interrupt request when buffer is empty.
  - Data sent at VDD or VregLogic levels selectable by mask option.

- **Prescaler**
  - 19 stages system clock divider down to 1Hz.
  - 3 interrupts requests: 9.4kHz; 586Hz and 1Hz.
  - Prescaler reset.

- **Supply voltage Level Detector**
  - 2 levels software selectable (2,2V or 2,5V).
  - Busy flag during measurement.
  - Interrupt request when measurement is ready.

- **10-Bit Universal Counter**
  - 10, 8, 6, 4 bit up/down counting.
  - 8 different input clocks.
  - Event counting with PA[0] and PA[3] as input clocks.
  - Full 10 bits or limited (8, 6, 4 bits) compare function.
  - 2 interrupt requests (on compare and on 0).
  - Pulse width modulation (PWM) output on PB[3].

- **Sleep Counter Reset (SCR)**
  - Wake up automatically the EM6640 from sleep.
  - 8 timings selectable by register.
  - Inhibit SCR by register.

- **Watchdogs**
  - Oscillation detection circuit.
  - Digital watchdog timer reset.

- **Interrupt Controller**
  - 4 external interrupt sources from PortA.
  - 5 internal interrupt sources: Prescaler (3), Timer (2), SVLD (1), EEPROM (1), SWB (1)

NB: <u>All frequencies written in this document are related to a typical system clock of 600 kHz.</u>

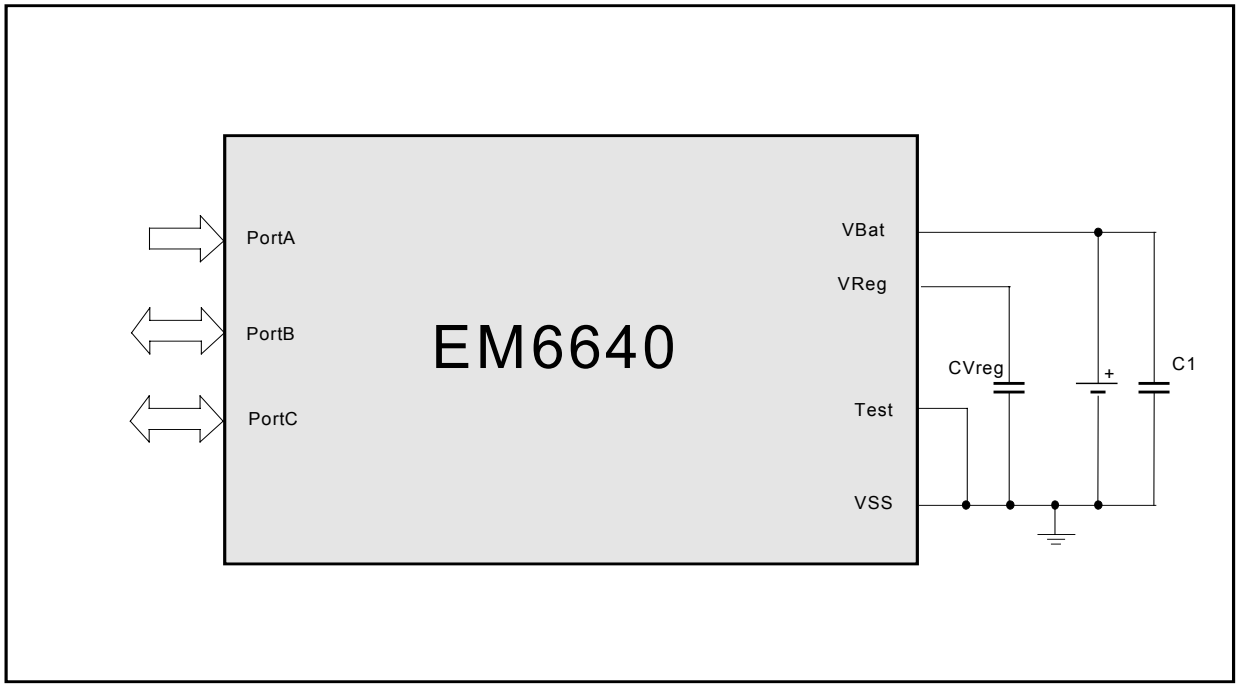
INDEX

1. Pin Description for EM6640 :

Pin Number	Pin Name	Function	Remarks
1	VReg	Internal voltage regulator	MFP programming connection

2	Test	Input test terminal	For EM test purpose only, GND 0 **!** And MFP programming connection
3	Port A[0]	Input port A[0]	testvar1, event counter input, IRQPA[0], SWB input clock
4	Port A[1]	Input port A[1]	testvar2, IRQPA[1]
5	Port A[2]	Input port A[2]	IRQPA[2]
6	Port A[3]	Input port A[3]	event counter input, IRQPA[3]
7	Port B[0]	Input/Output bitwise, cmos/open drain port B[0]	ck[20] output
8	Port B[1]	Input/Output bitwise, cmos/open drain port B[1]	ck[16] output
9	Port B[2]	Input/Output bitwise, cmos/open drain port B[2]	ck[12] output
10	Port B[3]	Input/Output bitwise, cmos/open drain port B[3]	PWM output
11	Port C[0]	Input/Output bitwise, cmos/open drain port C[0]	SWB Clock Out
12	Port C[1]	Input/Output bitwise, cmos/open drain port C[1]	SWB Data Out
13	Port C[2]	Input/Output bitwise, cmos/open drain port C[2]	
14	Port C[3]	Input/Output bitwise, cmos/open drain port C[3]	
15	VBat	Positive power supply terminal	VBat=VDD, MFP programming connection
16	VSS	Negative power supply terminal	reference terminal, MFP programming connection

Figure 3 **Typical configuration**



2. Operating modes

The EM6640 has two low power dissipation modes, STANDBY and SLEEP. Figure 4 is a transition diagram for these modes.

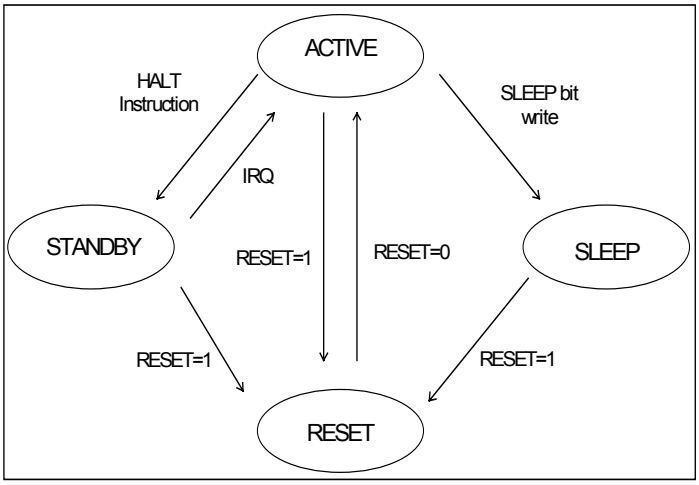
2.1 ACTIVE Mode

The active mode is the actual CPU running mode. Instructions are read from the internal ROM and executed by the CPU. Go into standby mode via the halt instruction or go into sleep mode by writing the sleep bit.

2.2 STANDBY Mode

Executing a HALT instruction puts the EM6640 into STANDBY mode. The voltage regulator, oscillator, watchdog timer, interrupts and timers/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 canceled by a RESET or an Interrupt request if enabled.

Figure 4 Mode transition diagram



2.3 SLEEP Mode

Writing the **Sleep** bit in the **RegSysCntl1** register puts the EM6640 in SLEEP mode. The oscillator stops and most functions of the EM6640 are inactive. To be able to write the **Sleep** bit, the **SleepEn** bit in **RegSysCntl2** must first be set to "1". In SLEEP mode only the voltage regulator is active. The RAM data integrity is maintained. SLEEP mode may be canceled only by the Input Reset from PortA or the Sleep Counter Reset.

During SLEEP mode and the following start up the EM6640 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 EM6640 was powered up (**SleepEn** = "0") or woken up from SLEEP mode (SleepEn = "1").

The bit **NoInputRes** in option register **OptPaRst** is inhibited is sleep mode.
TAKE CARE !!! To quit SLEEP mode, one must be sure to have a suitable defined combination of PortA inputs for reset (see section 4.3).

Table 2.3.1 shows the state of the EM6640 functions in STANDBY and SLEEP modes

FUNCTION	STANDBY	SLEEP
Oscillator	Active	Stopped
Oscillator Watchdog	Active	Stopped
Instruction Execution	Stopped	Stopped
Interrupt Functions	Active	Stopped
Registers and Flags	Retained	Reset
EEPROM	Retained	Retained
RAM data	Retained	Retained
Option Registers	Retained	Retained
Timer/Counter's	Active	Reset
Logic Watchdog	Active	Reset
Input PortA	Active **NoInputRes** = "0"	Active **NoInputRes** = "x"
I/O Port B	Active	High Impedance, no pulls
I/O Port C	Active	High Impedance, no pulls
SCR	Stopped	Active if enable
SWB	Active	Stopped
Voltage Level Detector	finishes on going measure, then stop	Stopped

3. Power Supply

The EM6640 is supplied by a single external power supply between VDD (VBat) and VSS (GND). A built-in voltage regulator generates VregLogic providing regulated voltage for the oscillator and the internal logic. An external capacitor (CVreg) has to be put on Vreg terminal (see Standard Operating Conditions, on page 55). Vreg terminal is not intended to be used with external load except CVreg.

The output drivers are supplied directly from the external supply VDD.

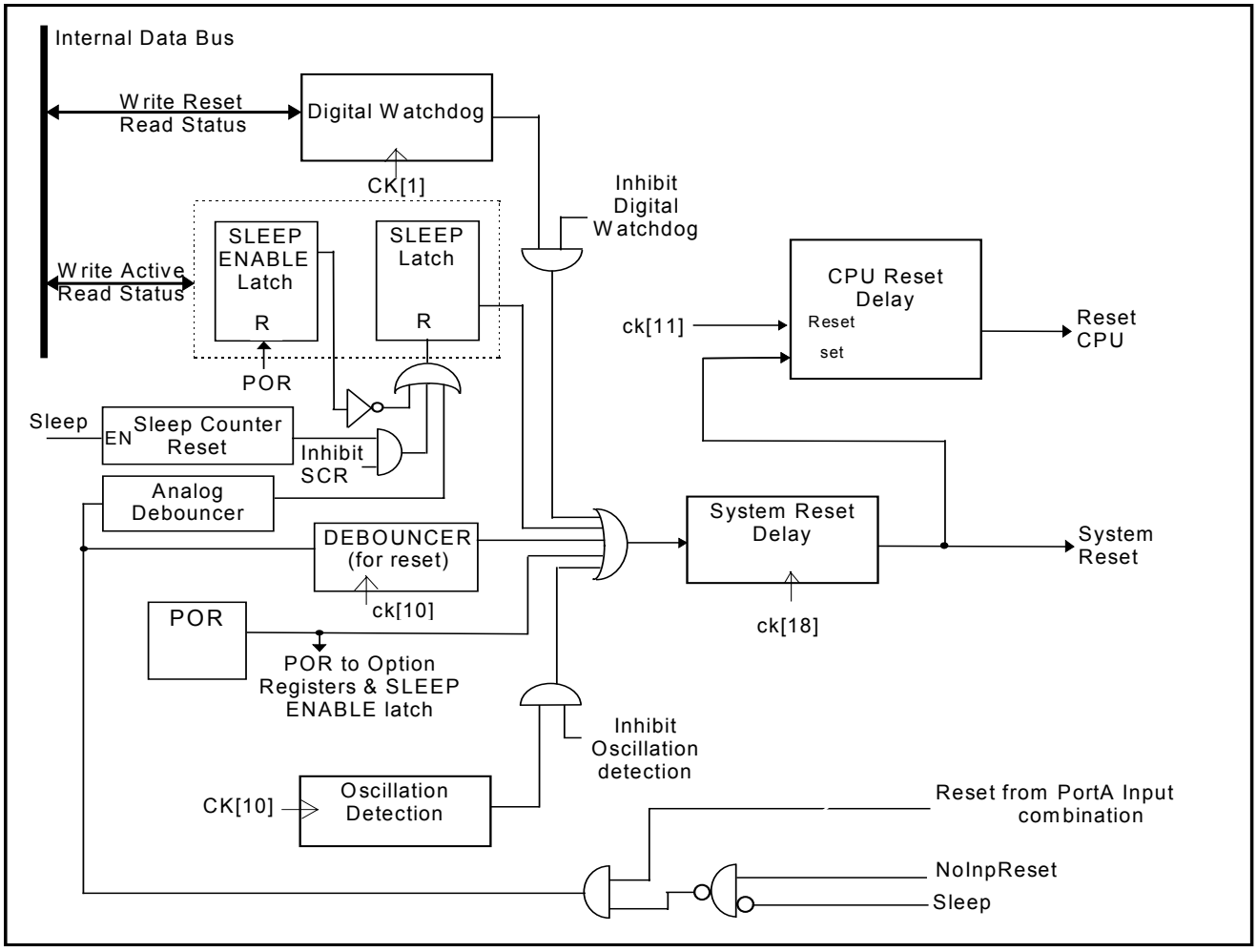
4. Reset

Figure 5 illustrates the reset structure of the EM6640. One can see that there are six possible reset sources:

(1) Internal initial reset from the Power On Reset (POR) circuitry.
(2) External reset by simultaneous high/low inputs to PortA.
(Combinations are defined in the registers **OptInpRSel1** and **OptInpRSel2**
(3) Internal reset from the Digital Watchdog.
(4) Internal reset from the Oscillation Detection Circuit.
(5) Internal reset when SLEEP mode is activated.
(6) Internal reset from Sleep Counter Reset.

All reset sources activate the System Reset and the Reset CPU. The 'System Reset Delay' ensures that the System Reset remains active long enough for all system functions to be reset (active for N SysClk cycles).
The 'CPU Reset Delay' ensures that the Reset CPU remains active until the oscillator is in stable oscillation.
As well as activating the System Reset and the Reset CPU, the POR also resets all Option Registers and the SLEEP ENABLE latch. System Reset and Reset CPU do not reset Option Registers nor the SLEEP ENABLE latch.

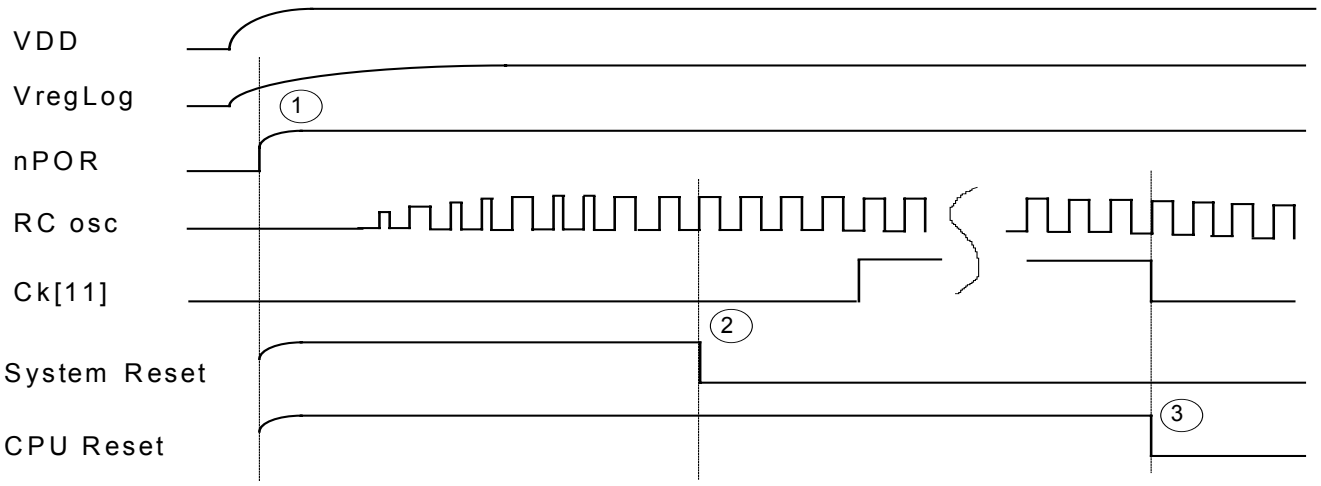
Figure 5. Reset structure



4.1 Power-Up

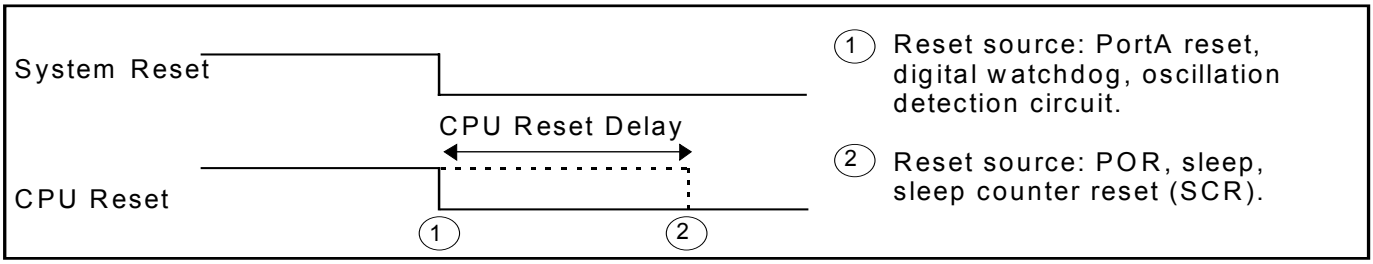
At power on, the voltage regulator starts to follow the supply voltage and triggers the power on reset circuitry, and thus the System Reset. The CPU of the EM6640 remains in the reset state for the '**CPU Reset Delay**', to allow the oscillator to stabilize after power up.
Because the oscillator is disabled during SLEEP mode, then when waking up from SLEEP mode, the CPU of the EM6640 remains in the reset state for the '**CPU Reset Delay'** also to allow the oscillator to stabilize.

The **'CPU Reset Delay'** is 512 system clocks (ck[11]) long.



1 When VregLog level reaches approx. 1.3v, the power on reset circuit removes nPOR.
2 The oscillator starts to run to supply a clock signal to the reset synchroniser and the prescaler blocks. After 16 oscillator clock cycles, the system Reset is removed.
3 After 512 oscillator clock cycles, there is a falling edge of ck[11] and the CPU Reset is removed.

Figure 6 Power On Startup



1 Reset source: PortA reset, digital watchdog, oscillation detection circuit.
2 Reset source: POR, sleep, sleep counter reset (SCR).

Figure 7 CPU Reset Delay (Coldstart)

4.2 Oscillation Detection Circuit

In ACTIVE or STANDBY modes, the Oscillator Detection Circuit monitors the oscillator. If it stops for any reason, a reset is generated for the '**CPU Reset Delay**'.
The oscillation detection circuitry can be inhibited with **NoOscWD** = 1 in register **RegSysCtl3.** At power up, and after any Reset, the function is activated.
During the CPU reset, the Oscillation Detection Circuit is inhibited because the oscillator is either off or in a stabilization period. Then, it will be active.
For the electrical specifications, see section : EM6640 Electrical specifications.

Table 4.2.1 Watchdog timer register **RegSysCntl3**

Bit	Name	Reset	R/W	Description
3	---			
2	---			
1	NoOscWD	0	R/W	No oscillator watchdog
0	*NoLogicWD*	*0*	*R/W*	*No logic watch dog*

4.3 Input-PortA-Reset

By writing the **OptInpRSel1** and **OptInpRSel2** registers it is possible to choose any combination of PortA input values to execute a system reset. The reset condition must be valid for at least 3.4mS (2 risings edges of ck[10] with system clock = 600kHz) in ACTIVE and STANDBY mode.
When canceling the SLEEP mode, the debouncer is not active (oscillator off). However, the reset condition passes through **an analogue filter**. The constant time is defined on the section EM6640 Electrical specifications, on page 56. In this case, the reset condition must be at least two times this constant time (In ACTIVE and STANDBY mode, the analogue filter is inhibited). The reset is generated for the 'CPU Reset Delay'

Bit **SelInpResMod** in option register **OptPaRst** selects which kind of input reset modes will be used: through an ″**OR logic**″ or an ″**AND logic**″ (default at power on). With the first mode, at least one input matching its reset condition will trigger a system reset. With the second mode, it is necessary to fully match the reset combination chosen in the registers.

Bit **NoInputReset** in option register **OptPaRst** selects the Input-PortA-Reset function in ACTIVE and STANDBY Mode. If set to "0" (default at power on) the occurrence of the selected combination for Input-PortA-Reset will trigger a system reset. Set to '1' the Input PortA Reset function is inhibited. This option bit has no action in SLEEP Mode, where the occurrence of the selected Input-PortA-Reset combination will always immediately trigger a system reset.

Reset combination selection (*InpReset)* in registers **OptInpRSel1** and **OptInpRSel2.**

SelinpResMod = 0 => *InpReset* = **InpResPA[0] • InpResPA[1] • InpResPA[2] • InpResPA[3]**
SelinpResMod = 1 => *InpReset* = **InpResPA[0] + InpResPA[1] + InpResPA[2] + InpResPA[3]**

SelinpResMod = 0

0InpRes1PA[n]	InpRes2PA[n]	InpResPA[n]
0	0	VSS
0	1	PA[n]
1	0	not PA[n]
1	1	VDD

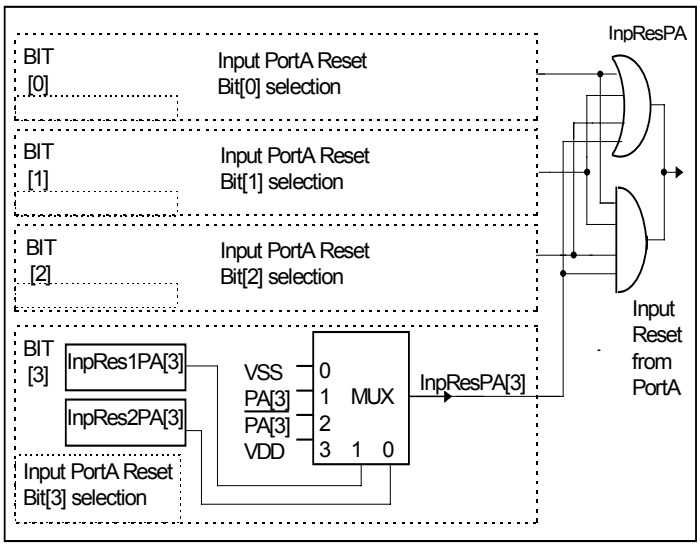
n = 0 to 3
i.e. ; - No reset if InpResPA[n] = VSS.
- With InpResPA[n] = VDD, concerning terminal [n] inhibited.
- Always Reset if InpResPA[3:0] = 'b1111.

SelinpResMod = 1

InpRes1PA[n]	InpRes2PA[n]	InpResPA[n]
0	0	VSS
0	1	PA[n]
1	0	not PA[n]
1	1	VDD

n = 0 to 3
i.e. ; - No reset if InpResPA[3:0] = 'b0000.
- With InpResPA[n] = VSS, concerning terminal [n] inhibited.
- Always Reset if InpResPA[n] = VDD.

Figure 8.Input PortA Reset structure



4.4 Sleep Counter Reset (SCR)

The Sleep Counter Reset, active only in SLEEP mode, will automatically trigger a reset which cancels Sleep Mode after a programmable time defined by writing bit **RstSlpTSel[2:0]** in option register **OptSlpCntRst**. The **EnSlpCntRst** bit in option register **OptSlpCntRst** set to "1" will enable the SCR as soon as the **SLEEP** bit in **RegSysCntl1** is written active. In this case, all the sleep considerations (low consumption) are taken into account (refer to chapter SLEEP Mode, page 5 for more details). When the SCR generates a reset, the CPU reset is removed after the 'CPU Reset Delay'.

The typical time constant of SCR is defined in the section EM6640 Electrical specifications, on page 55.
=> The minimum programmable time is 1x the time constant.
=> The maximum programmable time is 128x the time constant.

Figure 9 Sleep Counter Reset representation with SCR timing=2x as an example.

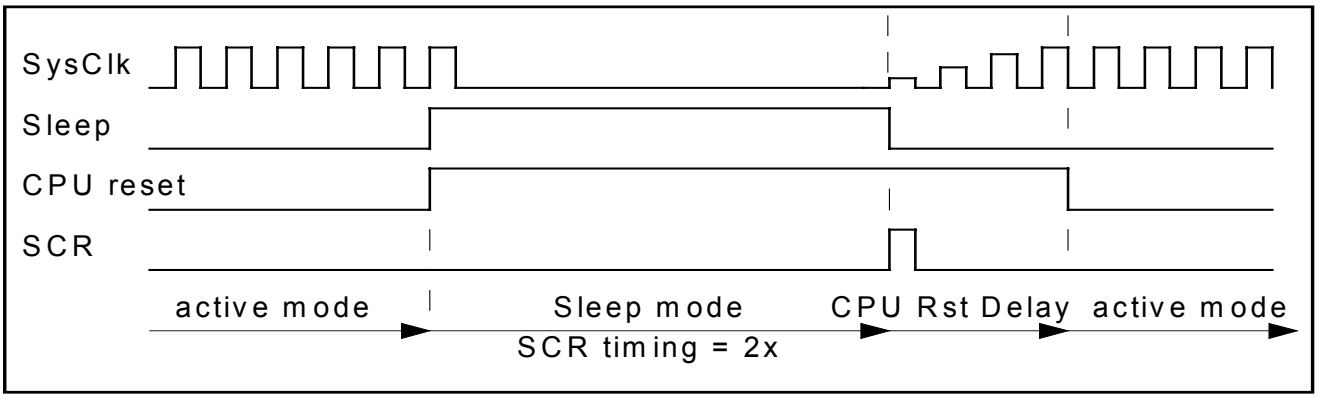


Table 10 register **OptSlpCntRst**

Bit	Name	Reset	R/W	Description
3	EnSlpCntRst	0	R/W	enable SCR
2	RstSlpTSel[2]	0	R/W	timing selection
1	RstSlpTSel[1]	0	R/W	timing selection
0	RstSlpTSel[0]	0	R/W	timing selection

Table 11 Sleep Counter Reset timing

RstSlpTSel[2]	RstSlpTSel[1]	RstSlpTSel[0]	SCR timing
0	0	0	1x
0	0	1	2x
0	1	0	4x
0	1	1	8x
1	0	0	16x
1	0	1	32x
1	1	0	64x
1	1	1	128x

4.5 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 (**NoLogicWD**) located in **RegSysCtl3**. By metal 1 mask option, one can force the Digital Watchdog to be always active, this mean that the bit **NoLogicWD** has no more effect (see: Digital Watchdog Option, page 50). In that case, the read of the bit **NoLogicWD** will always give '0'. By default, the Digital Watchdog can be controlled by the register **RegSysCtl3**. At power up, and after any System Reset, the watchdog timer is activated.

If for any reason the CPU stops, then the watchdog timer can detect this situation and activate 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 = 600kHz), 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 operates also in the 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 reset at any other time, then it could become active after just 2.5 seconds. In addition, using the Prescaler reset function can lower this minimum watchdog time. It is therefore recommended to use the Prescaler **IRQ1Hz** interrupt to periodically reset the watchdog every one 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]) : {WDVal1 WDVal0} '00', '01', '10', '11'. When in 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 its timer is reset and therefore always reads '0'.

Table 4.5.1 Watchdog timer register **RegSysCntl2**

Bit	Name	Reset	R/W	Description
3	WDReset	0	R/W	Reset the Watchdog 1 -> Resets the Logic Watchdog 0 -> no action The Read value is always '0'
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]

4.6 CPU State after Reset

Reset initializes the CPU as shown in Table 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 Oscillator

An RC oscillator generates the system operating clock for the CPU and peripheral circuits. The frequency can be adjusted if necessary by ~±32% in steps of ~1% by writing bits **OscAdj[5:0]** in the registers **OPTPaRST** and **OPTOscAdj**. The adjustable frequency range allowed is specified on page 55 (If you have a special request please contact EM Microelectronic Marin SA).
At power up, the default frequency is the lowest. The frequency is stored by adjusting the 6 bits in the EEPROM and transferring them to the registers **OPTPaRST** and **OPTOscAdj**. To increase frequency, put a higher calibration values and to decrease it, put a lower calibration values.
The adjustment value of the oscillator is written at EM-Marin at the last address of the EEPROM. By reading these 6 EEPROM bits and writing the contents to the registers **OPTOscAd**j and **OPTPaRST**, the delivery state will be 600kHz typical. See also section: EM6640 Electrical specifications, page 55.

Frequency adjustment procedure (example):
- 1st: selecting ck[20] output on PB[0] with the bit **PB600kHzOut** in register **OPTFSelPB.**
- 2nd: measure the output frequency with a frequency meter.
- 3rd: if it is not the desired frequency, modify the 6 bits dedicated to the RC oscillator in the registers **OPTOscAd**j and **OPTPaRST.**
- 4th: return to the point 2 until desired frequency is obtained.
- 5th: write the contents of the registers **OPTOscAd**j and **OPTPaRST** (2 MSB) to the EEPROM.
- (6th:read these 6 EEPROM bits and write the contents to the registers **OPTOscAd**j and **OPTPaRST**.)

The above procedure should be followed for the initial adjustment (first POR). For subsequent initializations the calibration values can be read from the EEPROM (start from point 6). It is not necessary to do this after any other resets (The values of the Option Registers are set by initial reset on power up and through write operations only).
To guarantee the good functionality of the whole circuit, it is recommended to adjust operating clock at 600 kHz.
User can decide himself which EEPROM address to use for the RC oscillator data if the last EEPROM address can not be kept.

Three different frequencies can be provided on the PortB[2:0] terminals (see section: PWM and Frequency output, on page 20). The highest is 600kHz which comes directly from the RC oscillator. The two others, 37.5kHz and 2.3kHz, come from the prescaler.

The oscillator circuit is supplied by the regulated voltage, VregLogic. In SLEEP mode the oscillator is stopped.
No external components are necessary.

Table 5.1.1 register **OPTPaRST**

Bit	Name	power on value	R/W	Description
3	OscAdj[5]	0	R/W	Adjustment of RC oscillator in EEPROM (MSB)
2	OscAdj[4]	0	R/W	Adjustment of RC oscillator in EEPROM
1	*SelinpResMod*	*0*	*R/W*	*input reset mode (Or or AND logic)*
0	*NoInputReset*	*0*	*R/W*	*PortA input reset option*

Default ″0″ is : no adjustment of the frequency

Table 5.1.2 register **OPTOscAdj**

Bit	Name	power on value	R/W	Description
3	OscAdj[3]	0	R/W	Adjustment of RC oscillator in EEPROM
2	OscAdj[2]	0	R/W	Adjustment of RC oscillator in EEPROM
1	OscAdj[1]	0	R/W	Adjustment of RC oscillator in EEPROM
0	OscAdj[0]	0	R/W	Adjustment of RC oscillator in EEPROM (LSB)

Default ″0″ is : no adjustment of the frequency

5.2 Prescaler

The input to the prescaler is the system clock signal. The prescaler consists of nineteen elements divider chain which delivers clock signals for the peripheral circuits such as timer/counter, debouncers and edge detectors, as well as generating prescaler interrupts. Power on initializes the prescaler to $0000.

Table 5.2.1 Prescaler clock name definition

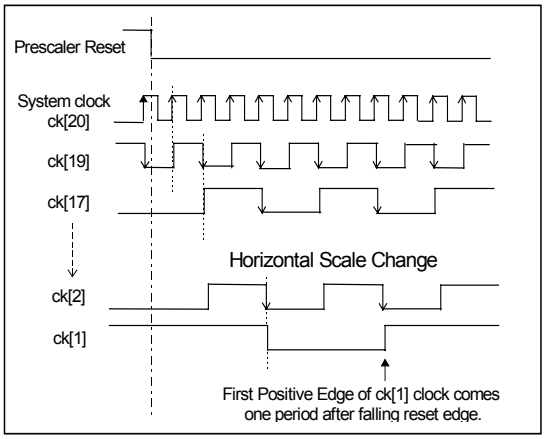
function	name	value
system clock	ck[20]	600000 Hz
system clock / 2	ck[19]	300000 Hz
system clock / 4	ck[18]	150000 Hz
system clock / 8	ck[17]	75000 Hz
system clock/ 16	ck[16]	37500 Hz
system clock / 32	ck[15]	18750 Hz
system clock / 64	ck[14]	9375 Hz
system clock / 128	ck [13]	4688 Hz
system clock / 256	ck [12]	2344 Hz
system clock / 512	ck[11]	1172 Hz

function	name	value
system clock / 1024	ck[10]	586 Hz
system clock / 2048	ck[9]	293 Hz
system clock / 4096	ck[8]	146 Hz
system clock / 8192	ck[7]	73 Hz
system clock / 16384	ck[6]	37 Hz
system clock / 32768	ck[5]	18 Hz
system clock / 65536	ck[4]	9.2 Hz
system clock / 131072	ck[3]	4.6 Hz
system clock / 262144	ck[2]	2.3 Hz
system clock / 524288	ck[1]	1.1 Hz

Table 5.2.2 Control of prescaler register **RegPresc**

Bit	Name	Reset	R/W	Description
3	*PWMOn*	*0*	*R/W*	*see 10 bit counter*
2	ResPresc	0	R/W	Write Reset prescaler 1 -> Resets the divider chain from ck[17] down to ck[1] 0 -> no action. The Read value is always '0'
1		0	R	The Read value is always '0'
0	DebSel	0	R/W	Debouncer clock select. 0 -> debouncer with ck[8] 1 -> debouncer with ck[17]

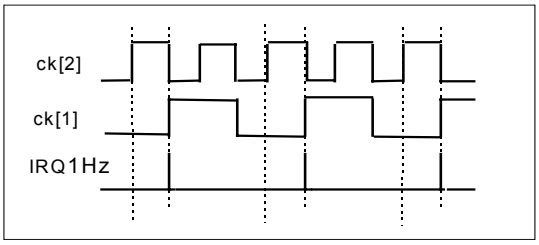
Figure 12. Prescaler frequency timing



The Prescaler contains 3 interrupt sources:
- IRQ9k4Hz ; this is ck[14] positive edge interrupt.
- IRQ586Hz ; this is ck[10] positive edge interrupt.
- IRQ1Hz ; this is ck[1] positive edge interrupt.

There is no interrupt generation on reset.
In reset mode, ck[1] is set to a high level.
The first IRQ1Hz Interrupt occurs ~1sec (600kHz) after reset.

Figure 13. Prescaler Interrupts example:



6. Input and Output ports

The EM6640 has:

- one 4-bit input port (PortA)
- two 4-bit input/output ports. (PortB & PortC)

6.1 Ports overview

Table 6.1.1 Input and Output ports overview

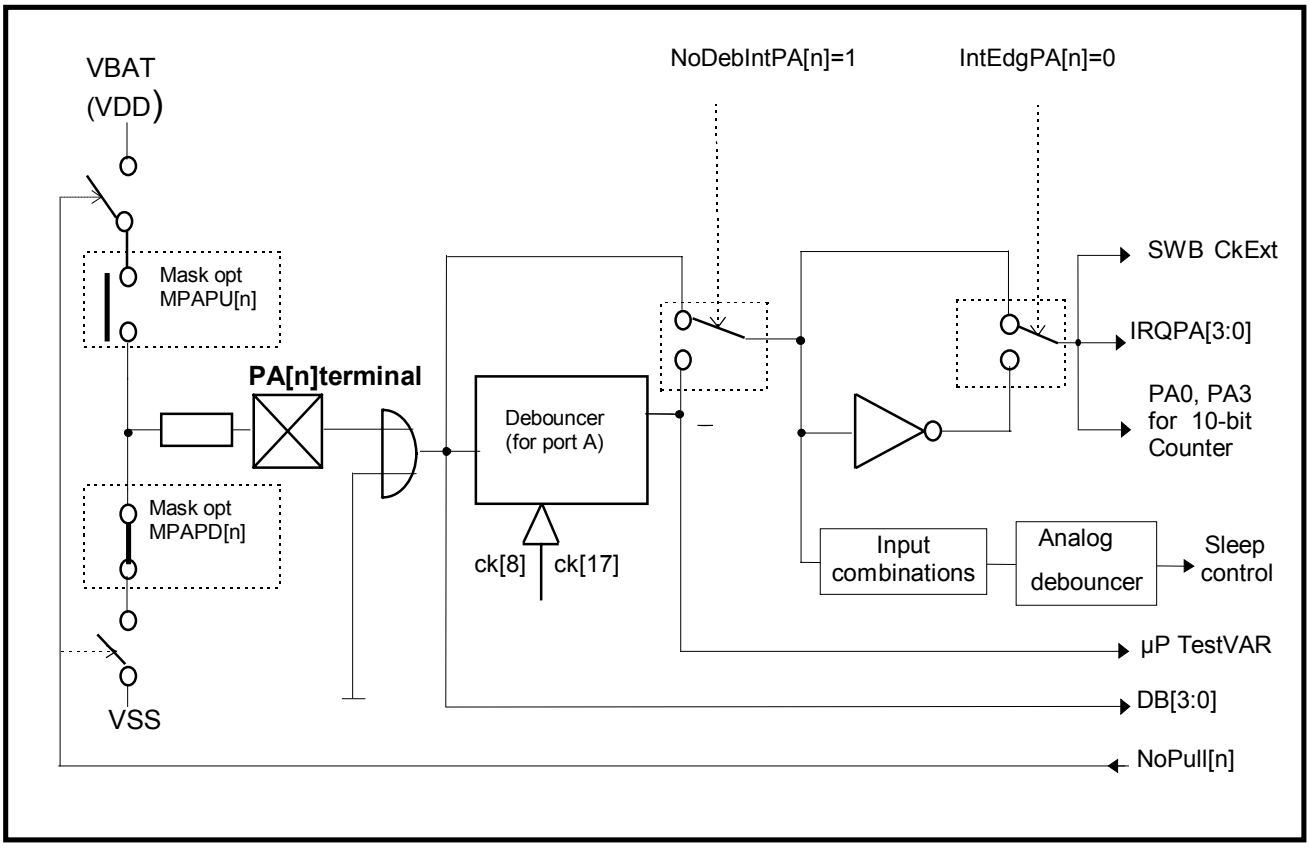
Port	Mode	Mask(M:) or Register(R:) option	Function	Bitwise Multi Function on Ports			
PA [3:0]	Input	M: Pull-up M: Pull-down (default) R: Pull(up/down) select R: Debounced or direct input for IRQ request and Counter R: + or - for IRQ-edge and Counter R: Input reset combination	-Input -Bitwise Interrupt request -PA[3],PA[0] input for the Event Counter -Software Test Variable conditional jump -PortA Reset inputs -SWB external input clock	**PA[3]** IRQPA3 10 bit Event Counter clock	**PA[2]** IRQPA2 -	**PA[1]** IRQPA1 - Test Var2	**PA[0]** IRQPA0 10 bit Event Counter clock Test Var1 SWB CkExt
PB [3:0]	bitwise input or output	R: CMOS or Nch open drain output R: Pull-Down on input R: Pull-Up on input	-Input or Output -PB[3] for the PWM output -PB[2:0] for the ck[20,16, 12] output -Tristatable in sleep mode	**PB[3]** PWM output	**PB[2]** ck[12] output	**PB[1]** ck[16] output	**PB[0]** ck[20] output
PC [3:0]	bitwise input or output	R: CMOS or Nch open drain output R: Pull-Down on input R: Pull-Up on input	-Input or Output -PC[1:0] for the SWB -Tristatable in sleep mode	**PC[3]**	**PC[2]**	**PC[1]** SWB Data out	**PC[0]** SWB Clock out

6.2 PortA

The EM6640 has one four bit general purpose CMOS input port. The PortA input can be read at any time, pull-up or pull-down resistors can be chosen by metal mask. All selections concerning PortA are bitwise executable. I.e. Pull-up on PA[2], pull-down on PA[0], positive IRQ edge on PA[0] but negative on PA[1], etc.

In SLEEP mode the PortA inputs are continuously monitored to match the input reset condition which will immediately wake the EM6640 from SLEEP mode. The pull-up or pull-down resistors remain active as defined in the option register.

Figure 14. Input PortA configuration



6.2.1 IRQ on portA

For interrupt request generation (IRQ) one can choose direct or debounced input and positive or negative edge IRQ triggering. With the debouncer selected (**OPtDebIntPA**) the input must be stable for two rising edges of the selected debouncer clock ck[8] (default) or ck[17] (**RegPresc**), this means a worst case of 14mS (default) or 27µs with a system clock of 600kHz.
Either a positive or a negative edge on the PortA inputs - after debouncer or not - can generate an interrupt request. This selection is done in the option register **OPTIntEdgPA.**

All four bits of PortA can provide an IRQ, each pin with its own interrupt mask bit in the **RegIRQMask1** register. When an IRQ occurs, inspection of the **RegIRQ1**, **RegIRQ2** and **RegIRQ3** 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 10.

6.2.2 Pull-up/down

Each of the input port terminals PA[3:0] has a resistor integrated which can be used either as pull-up or pull-down resistor, depending on the selected metal mask options. See Table 6.2.1 and the PortA metal mask chapter for details. The pull resistor can be inhibited using the **NoPullPA[n]** bits in the register **OptNoPullPA.**

Table 6.2.1 Pull-up or pull-down resistor on PortA select

opt mask pull-up MPAPU[n]	opt mask pull-down MPAPD[n]	NoPullPA[n] value	Action
no	no	x	no pull-up, no pull-down
no	yes	0	no pull-up, pull-down
no	yes	1	no pull-up, no pull-down
yes	no	0	pull-up, no pull-down
yes	no	1	no pull-up , no pull-down
yes	yes	x	not allowed*

with n=0...3

* only pullup or pulldown may be chosen on any PortA terminal (one choice is excluding the other)

Any PortA 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.

6.2.3 Software test variables

The PortA terminals PA[3:0] are also used as input conditions for conditional software branches. Independent of the **OPtDebIntPA** and the **OPTIntEdgPA** these CPU inputs are **always** debounced and non-inverted.

- debounced PA[0] is connected to CPU TestVar1
- debounced PA[1] is connected to CPU TestVar2

6.2.4 PortA for 10-bit Counter

The PA[0] and PA[3] inputs can be used as the clock input terminal for the 10 bit counter in "event count" mode. As for the IRQ generation one can choose debounced or direct input with the register **OPtDebIntPA** and non-inverted or inverted input with the register **OPTIntEdgPA**. Debouncer input is recommended when using PA[3] or PA[0] for the event counting.

6.2.5 PortA for serial write buffer (SWB)

The PA[0] can be used as the external clock input terminal for the SWB in automatic mode. Depending of the register **RegSWBCntl2** contents, this external clock can be divided by 1/1, 1/4, 1/88 or 1/352.
As for the IRQ generation one can choose debounced or direct input with the register **OPtDebIntPA** and non-inverted or inverted input with the register **OPTIntEdgPA**.

6.3 PortA registers

Table 6.3.1 register **RegPA**

Bit	Name	Reset	R/W	Description
3	PAData[3]	-	R	PA[3] input status
2	PAData[2]	-	R	PA[2] input status
1	PAData[1]	-	R	PA[1] input status
0	PAData[0]	-	R	PA[0] input status

direct read on pin

Table 6.3.2 register **RegIRQMask1**

Bit	Name	Reset	R/W	Description
3	MaskIRQPA[3]	0	R/W	interrupt mask for PA[3] input
2	MaskIRQPA[2]	0	R/W	interrupt mask for PA[2] input
1	MaskIRQPA[1]	0	R/W	interrupt mask for PA[1] input
0	MaskIRQPA[0]	0	R/W	interrupt mask for PA[0] input

Default "0" is: interrupt request masked, no new request stored

Table 6.3.3 register **RegIRQ1**

Bit	Name	Reset	R/W	Description
3	IRQPA[3]	0	R/W*	interrupt request on PA[3]
2	IRQPA[2]	0	R/W*	interrupt request on PA[2]
1	IRQPA[1]	0	R/W*	interrupt request on PA[1]
0	IRQPA[0]	0	R/W*	interrupt request on PA[0]

W*; Write "1" clears the bit, write "0" has no action, Default "0" is: No Interrupt request

Table 6.3.4 register **OPTIntEdgPA**

Bit	Name	power on value	R/W	Description
3	IntEdgPA[3]	0	R/W	interrupt edge select for PA[3]
2	IntEdgPA[2]	0	R/W	interrupt edge select for PA[2]
1	IntEdgPA[1]	0	R/W	interrupt edge select for PA[1]
0	IntEdgPA[0]	0	R/W	interrupt edge select for PA[0]

Default "0" is: Positive edge selection

Table 6.3.5 register **OPTDebIntPA**

Bit	Name	power on value	R/W	Description
3	NoDebIntPA[3]	0	R/W	interrupt debounced for PA[3]
2	NoDebIntPA[2]	0	R/W	interrupt debounced for PA[2]
1	NoDebIntPA[1]	0	R/W	interrupt debounced for PA[1]
0	NoDebIntPA[0]	0	R/W	interrupt debounced for PA[0]

Default "0" is: Debounced inputs for interrupt generation

Table 6.3.6 register **OPTNoPullPA**

Bit	Name	power on value	R/W	Description
3	NoPullPA[3]	0	R/W	pull-up/down selection on PA[3]
2	NoPullPA[2]	0	R/W	pull-up/down selection on PA[2]
1	NoPullPA[1]	0	R/W	pull-up/down selection on PA[1]
0	NoPullPA[0]	0	R/W	pull-up/down selection on PA[0]

Default "0" is: see Table 6.2.1 Pull-up or pull-down resistor on PortA select

Table 6.3.7 Register **OPTPaRST**

Bit	Name	power on value	R/W	Description
3	*OscAdj[5]*	***1***	*R/W*	*Adjustment of RC oscillator in EEPROM (MSB)*
2	*OscAdj[4]*	*0*	*R/W*	*Adjustment of RC oscillator in EEPROM*
1	SelinpResMod	0	R/W	PortA input reset option
0	NoInputReset	0	R/W	input reset mode (Or or AND logic)

Default ″0″ is : bit[0] PortA can reset the EM6640, bit[1] input reset mode: AND logic

6.4 PortB

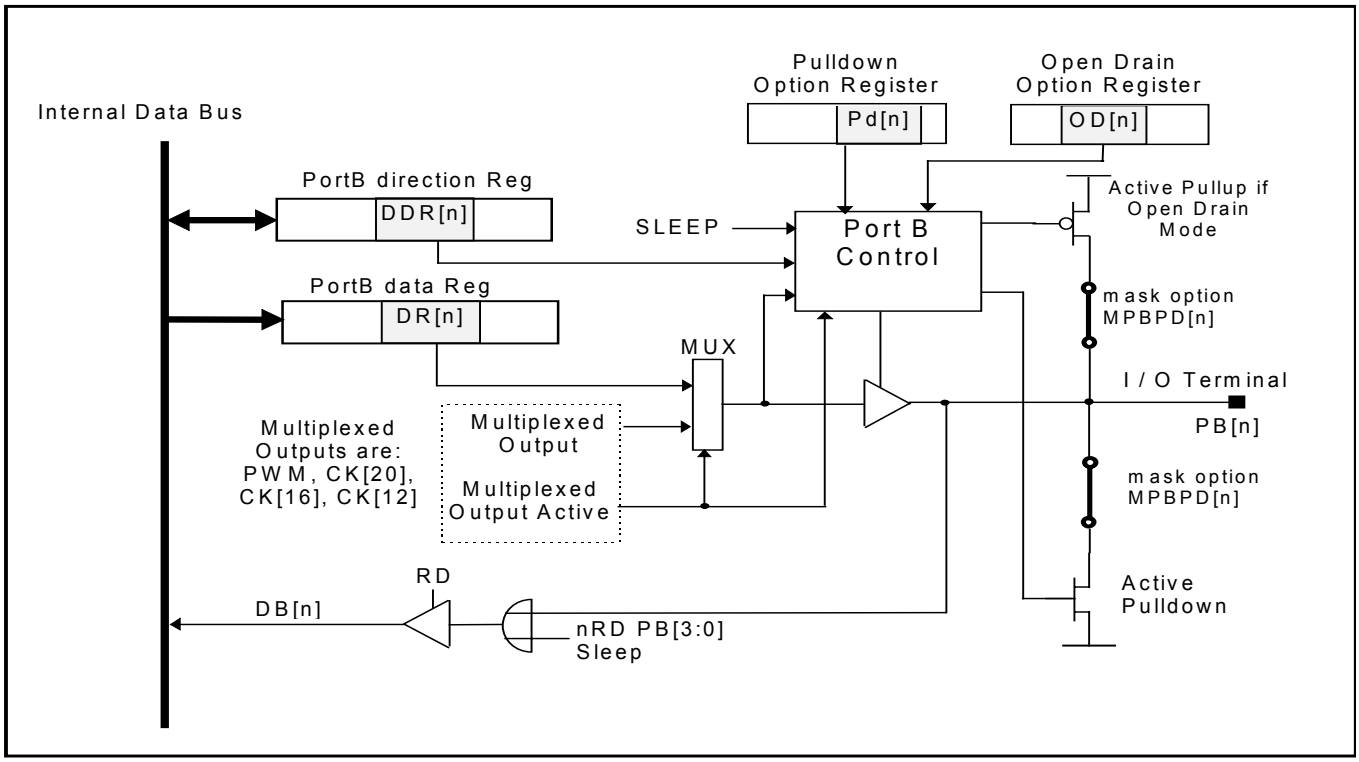
The EM6640 has two four bit general purpose I/O ports: PortB and PortC. Each bit can be configured individually (bitwise port) by software for input/output, pull-up, pull-down, CMOS/Nchannel Open Drain output, Frequency or PWM output. The PortB has two high current output pads: PB[2:1].

6.4.1 Input / Output Mode

Each PortB terminal can be either input or output. These modes can be set by writing the corresponding bit in the **RegPBCntl** control register. The **RegPBData** register contains the data written to the PortB terminal in output mode. To set for input (default), 0 is written to the corresponding bit of the **RegPBCntl** register which results in a high impedance state for the output driver. The output mode is set by writing 1 in the control register, and consequently the output terminal follows the status of the bits in the **RegPBData** register. The PortB terminal status can be read in any mode (read at address of **RegPBData**).

During SLEEP and Reset mode, PB[3:0] is in high impedance state with no pulls up/down active. Except during a read phase on PortB (in active mode), all the PortB inputs are cut off (blocked).

Figure 15. PortB architecture



6.4.2 Pull-up/Down

For each terminal of PB[3:0] an input pull-up (metal mask MPBPU[n]) or pull-down (metal mask MPBPD[n]) resistor can be implemented per metal mask option. Per default the two metal masks are in place, so one can chose per software to have either a pull-up, a pull-down or no resistor.
For Metal mask selection and available resistor values refer to chapter 'PortB Metal Options'.

Pulldown ON : MPBPD[n] must be in place ,
AND the bit **NoPdPB**[n] must be '0' .

Pulldown OFF : MPBPD[n] is not in place,
OR if MPBPD[n] is in place **NoPdPB**[n] = '1' cuts off the pulldown.
OR selecting **NchOpDPB**[n] = '1' cuts off the pulldown.

Pullup ON : MPBPU[n] must be in place,
AND the bit **NchOpDPB**[n] must be '1' ,
AND the bit **PBIOCntl[n]** = '0' (input mode) **OR** if **PBIOCntl[n]** = '1' while **PBData**[n] = 1.

Pullup OFF : MPBPU[n] is not in place,
OR if MPBPU[n] is in place **NchOpDPB**[n] = '0' cuts off the pullup,
OR if MPBPU[n] is in place and if **NchOpDPB**[n] = '1' then **PBData**[n] = 0 cuts off the pullup.

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

For **POWER SAVING** one can switch off the PortB pull resistors between two read phases. No cross current flows in the input amplifier while the PortB is not read. The recommended order is :

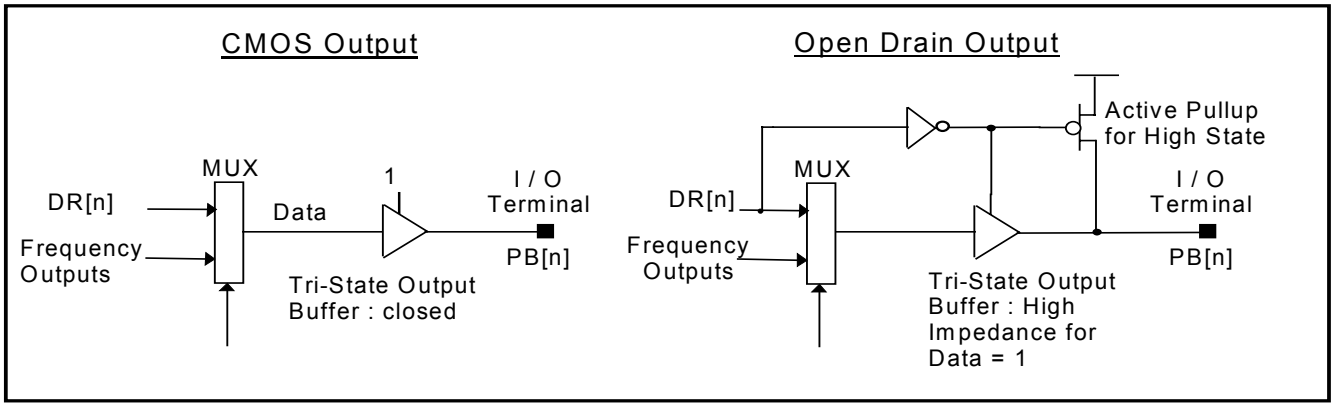
- switch on the pull resistor.
- allow sufficient time - RC constant - for the pull resistor to drive the line to either VSS or VDD.
- Read the PortB
- Switch off the pull resistor

Minimum time with current on the pull resistor is 4 periods of the system clock, if the RC constant is lower than 1 system clock period. Adding a NOP before reading moves the number of periods with current in the pull resistor to 6 and the maximum RC delay to 3 clock periods.

6.4.3 CMOS / Nchannel Open Drain Output

The PortB outputs can be configured as either CMOS or Nchannel Open Drain outputs. In CMOS both logic '1' and '0' are driven out on the terminal. In Nchannel Open Drain only the logic '0' is driven out on the terminal, the logic '1' value is defined by the pull-up resistor (if existing).

Figure 16. CMOS or Open Drain outputs



6.4.4 PWM and Frequency output

PB[3] can also be used to output the PWM (Pulse Width Modulation) signal from the 10-Bit Counter (refer to 10-bit Counter chapter).
PB[2:0] can be used to output three different frequencies from the RC oscillator: ck[20], ck[16] and ck[12].

-Selecting ck[20] output on PB[0] with bit **PB600kHzOut** in register **OPTFSelPB.**
-Selecting ck[16] output on PB[1] with bit **PB37k5HzOut** in register **OPTFSelPB.**
-Selecting ck[12] output on PB[2] with bit **PB2k3HzOut** in register **OPTFSelPB.**
-Selecting PWM output on PB[3] with bit **PWMOn** in register **RegPresc.**

6.5 PortB registers

Table 6.5.1 register **RegPBData**

Bit	Name	Reset	R/W	Description
3	PBData[3]	-	R* /W	PB[3] input and output
2	PBData[2]	-	R* /W	PB[2] input and output
1	PBData[1]	-	R* /W	PB[1] input and output
0	PBData[0]	-	R* /W	PB[0] input and output

R* : direct read on pin (not the internal register read).

Table 6.5.2 register **RegPBCntl**

Bit	Name	Reset	R/W	Description
3	PBIOCntl[3]	0	R/W	I/O control for PB[3]
2	PBIOCntl[2]	0	R/W	I/O control for PB[2]
1	PBIOCntl[1]	0	R/W	I/O control for PB[1]
0	PBIOCntl[0]	0	R/W	I/O control for PB[0]

Default "0" is: PortB in input mode

Table 6.5.3 register **OPTFSelPB**

Bit	Name	power on value	R/W	Description
3	-	-	-	-
2	PB2k3HzOut	0	R/W	ck[12] output on PB[2]
1	PB37k5HzOut	0	R/W	ck[16] output on PB[1]
0	PB600kHzOut	0	R/W	ck[20] output on PB[0]

Default "0" is: No frequency output.

Table 6.5.4 option register **OPTNoPdPB**

Bit	Name	power on value	R/W	Description
3	NoPdPB[3]	0	R/W	No pull-down on PB[3]
2	NoPdPB[2]	0	R/W	No pull-down on PB[2]
1	NoPdPB[1]	0	R/W	No pull-down on PB[1]
0	NoPdPB[0]	0	R/W	No pull-down on PB[0]

Default "0" is: Pull-down on

Table 6.5.5 option register **OPTNchOpDPB**

Bit	Name	power on value	R/W	Description
3	NchOpDPB[3]	0	R/W	N-Channel Open Drain on PB[3]
2	NchOpDPB[2]	0	R/W	N-Channel Open Drain on PB[2]
1	NchOpDPB[1]	0	R/W	N-Channel Open Drain on PB[1]
0	NchOpDPB[0]	0	R/W	N-Channel Open Drain on PB[0]

Default "0" is: CMOS on PB[3..0]

6.6 PortC

PortC has the same features as the PortB but instead of being used to outputs frequencies, it can be used to output Serial Write Buffer (SWB) signals.

6.6.1 Input / Output Mode

Each PortC terminal can be either input or output. These modes can be set by writing the corresponding bit in the **RegPCCntl** control register. The **RegPCData** register contains the data written to the PortC terminal in output mode. To set for input (default), 0 is written to the corresponding bit of the **RegPCCntl** register which results in a high impedance state for the output driver. The output mode is set by writing 1 in the control register, and consequently the output terminal follows the status of the bits in the **RegPCData** register. The PortC terminal status can be read in any mode (read at address of **RegPCData**).

During SLEEP and Reset mode, PC[3:0] is in high impedance state with no pulls up/down active. Except during a read phase on PortC (in active mode), all the PortC inputs are cut off (blocked).

6.6.2 Pull-up/Down

For each terminal of PC[3:0] an input pull-up (metal mask MPCPU[n]) or pull-down (metal mask MPCPD[n]) resistor can be implemented per metal mask option. Per default the two metal masks are in place, so one can chose per software to have either a pull-up, a pull-down or no resistor.

For Metal mask selection and available resistor values refer to chapter 'PortC Metal Options'.

Pulldown ON : MPCPD[n] must be in place ,
AND the bit **NoPdPC**[n] must be '0' .

Pulldown OFF : MPBPD[n] is not in place,
OR if MPCPD[n] is in place **NoPdPC**[n] = '1' cuts off the pulldown.
OR seletcing **NchOpDPC**[n] = '1' cuts off the pulldown.

Pullup ON : MPCPU[n] must be in place,
AND the bit **NchOpDPC**[n] must be '1' ,
AND the bit **PCIOCntl[n]** = '0' (input mode) **OR** if **PCIOCntl[n]** = '1' while **PCData**[n] = 1.

Pullup OFF : MPCPU[n] is not in place,
OR if MPCPU[n] is in place **NchOpDPC**[n] = '0' cuts off the pullup,
OR if MPCPU[n] is in place and if **NchOpDPC**[n] = '1' then **PCData**[n] = 0 cuts off the pullup.

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

For **POWER SAVING** one can switch off the PortC pull resistors between two read phases. No cross current flows in the input amplifier while the PortC is not read. The recommended order is :

- switch on the pull resistor.
- allow sufficient time - RC constant - for the pull resistor to drive the line to either VSS or VDD.
- Read the PortC
- Switch off the pull resistor

Minimum time with current on the pull resistor is 4 periods of the system clock, if the RC constant is lower than 1 system clock period. Adding a NOP before reading moves the number of periods with current in the pull resistor to 6 and the maximum RC delay to 3 clock periods.

6.6.3 CMOS / Nchannel Open Drain Output

The PortC outputs can be configured as either CMOS or Nchannel Open Drain outputs. In CMOS both logic '1' and '0' are driven out on the terminal. In Nchannel Open Drain only the logic '0' is driven out on the terminal, the logic '1' value is defined by the pull-up resistor (if existing).

6.6.4 Serial Write Buffer (SWB)

If the serial write buffer is actived by **EnSWB** bit in **RegSWBCntl**, PC[0] is the output terminal for the serial clock and PC[1] is the output terminal for the serial data. For details, see section: 8 Serial (Output) Write Buffer - SWB.

6.7 PortC registers

Table 6.7.1 register **RegPCData**

Bit	Name	Reset	R/W	Description
3	PCData[3]	-	R/W*	PC[3] input and output
2	PCData[2]	-	R/W*	PC[2] input and output
1	PCData[1]	-	R/W*	PC[1] input and output
0	PCData[0]	-	R/W*	PC[0] input and output

R* : direct read on pin (not the internal register read).

Table 6.7.2 register **RegPCCntl**

Bit	Name	Reset	R/W	Description
3	PCIOCntl[3]	0	R/W	I/O control for PC[3]
2	PCIOCntl[2]	0	R/W	I/O control for PC[2]
1	PCIOCntl[1]	0	R/W	I/O control for PC[1]
0	PCIOCntl[0]	0	R/W	I/O control for PC[0]

Default "0" is : PortC in input mode

Table 6.7.3 option register **OPTNoPdPC**

Bit	Name	power on value	R/W	Description
3	NoPdPC[3]	0	R/W	No pull-down on PC[3]
2	NoPdPC[2]	0	R/W	No pull-down on PC[2]
1	NoPdPC[1]	0	R/W	No pull-down on PC[1]
0	NoPdPC[0]	0	R/W	No pull-down on PC[0]

Default "0" is: Pull-down on

Table 6.7.4 option register **OPTNchOpDPC**

Bit	Name	power on value	R/W	Description
3	NchOpDPC[3]	0	R/W	N-Channel Open Drain on PC[3]
2	NchOpDPC[2]	0	R/W	N-Channel Open Drain on PC[2]
1	NchOpDPC[1]	0	R/W	N-Channel Open Drain on PC[1]
0	NchOpDPC[0]	0	R/W	N-Channel Open Drain on PC[0]

Default "0" is: CMOS on PC[3..0]

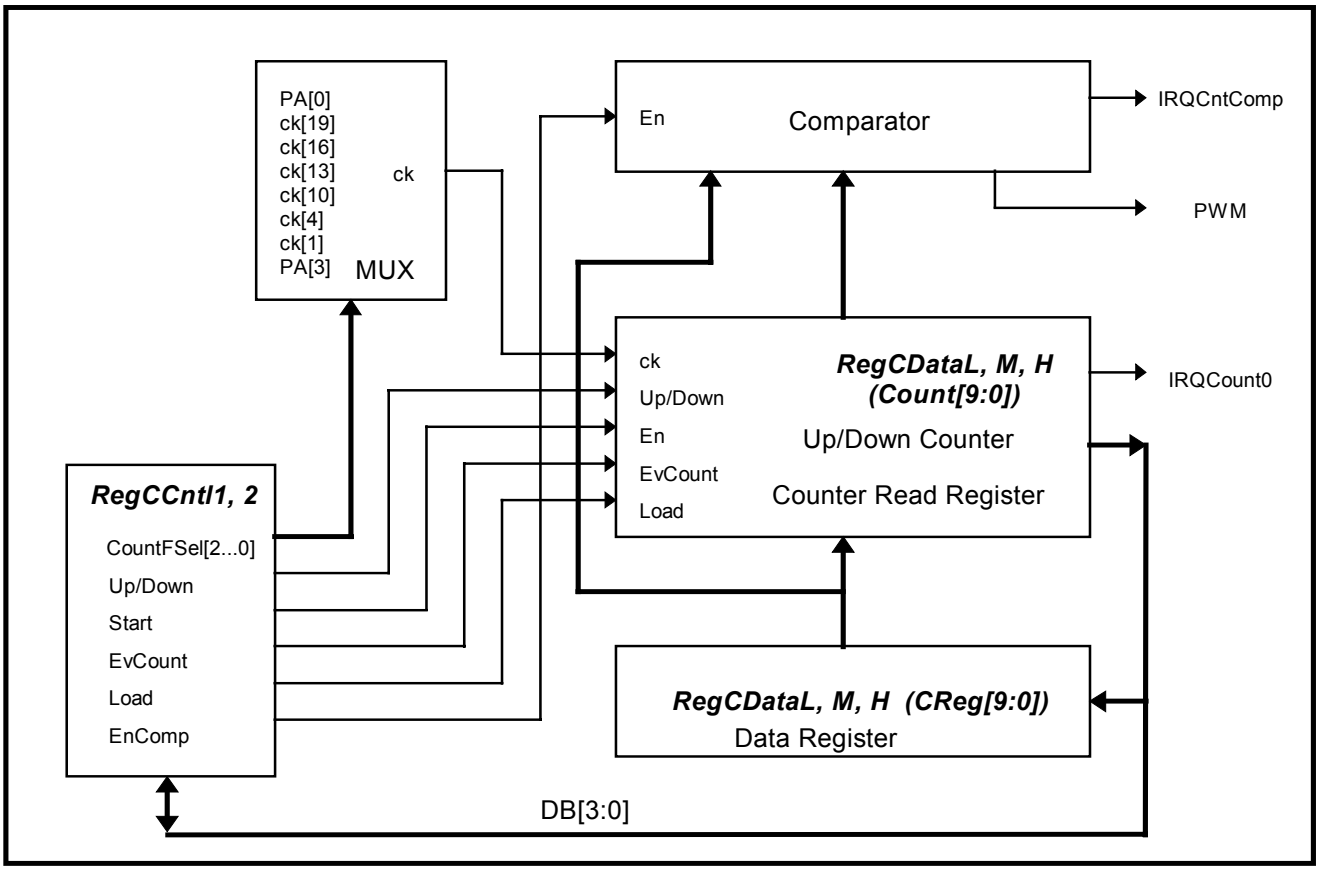
7. 10-bit Counter

The EM6640 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 of these clocks are coming from the input pads PA[0] and PA[3]. 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 matched the compare data register value. Each of this interrupt requests can be masked (default). See section 10 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 can be generated and output on PortB PB[3].

Figure 17. 10-bit Counter Block Diagram



7.1 Full, 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 **SelIntComp.** Refer to chapter 7.4.

Table 6.7—1. Counter length selection

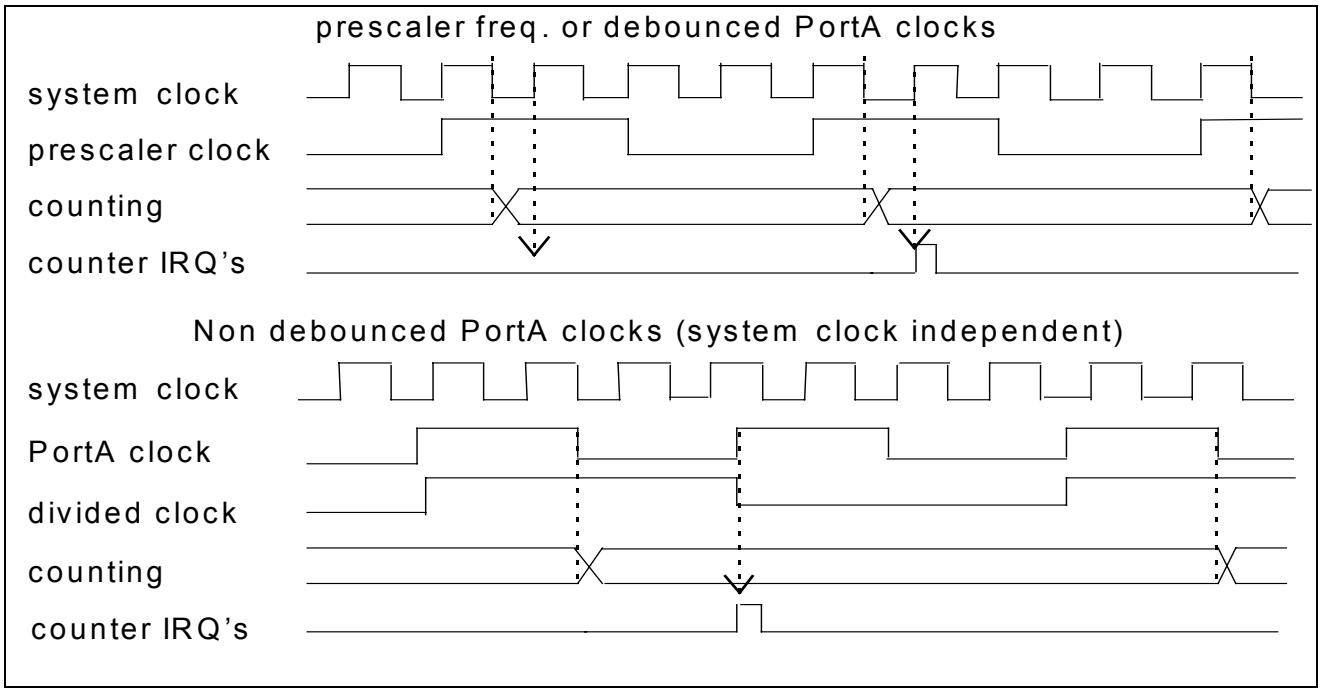
BitSel[1]	BitSel[0]	counter length
0	0	10-Bit
0	1	8-Bit
1	0	6-Bit
1	1	4-Bit

7.2 Frequency Select and Up/Down Counting

8 different input clocks can be selected to drive the Counter. The selection is done with bits **CountFSel2…0** in register **RegCCntl1**. 6 of this input clocks are coming from the prescaler. The maximum prescaler clock frequency for the counter is half the system clock and the lowest is 1Hz. Therefore a complete counter roll over can take as much as 17.07min (1Hz clock, 10 bit length) or as little as 53.3µs (ck[19], 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[20]) rising edge. IRQCount0 condition in UpCount Mode is : reaching 3FF if 10-bit counter length (resp FF, 3F, F in 8, 6, 4-bit counter length). In DownCount Mode the condition is reaching '0'. The nonselected bits are 'don't care'. For IRQComp refer to section 7.4.

Note: The Prescaler and the Microprocessor clock's are usually non-synchronous, therefore timebases 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 the µP commands using the prescaler reset function.

Figure 18. Counter Clock Timing



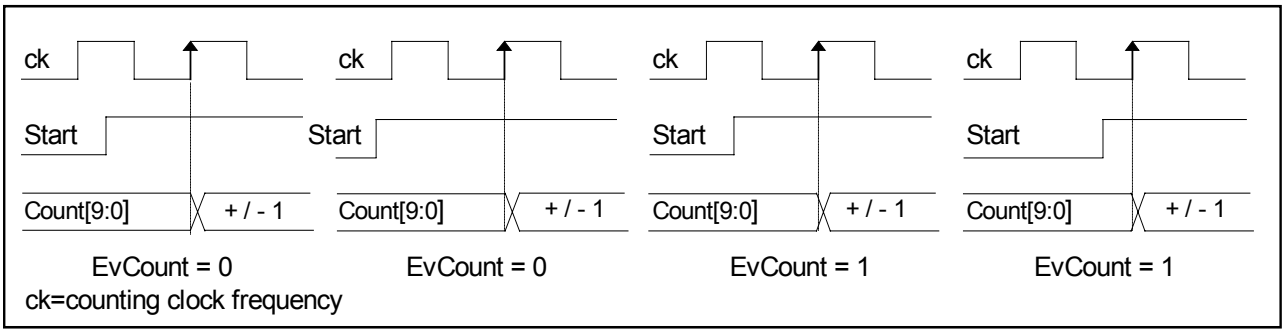
The two remaining clock sources are coming from the PA[0] or PA[3] terminals. Refer to chapter PortA on page 16 for details. Both sources can be either debounced (ck[17], ck[8]) or direct inputs, the input polarity can also be chosen. The output after the debouncer polarity selector is named PA3 , PA0 resp. For the debouncer and input polarity selection, refer to chapter 6.2.1 on page 16.
In the case of PortA input clock without debouncer, the counting clock frequency will be half the input clock on PortA. The counter advances on every odd numbered PortA negative edge (divided clock is high level). IRQCount0 and IRQComp will be generated on the rising PA3 or PA0 input clock edge. In this condition the EM6640 is able to count with a higher clock rate as its internal system clock (Hi-Frequency Input).
(Maximum PortA input frequency is at least 1MHz (@VDD ≥ 1.9V)).

In both, up or down count mode, the counter is cyclic. The counting direction is chosen in register **RegCCntl1** bit **Up/Down** (default=0, down counting). 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 19. Internal clock synchronization.

7.3 Event Counting

The counter can be used in a special event count mode where a certain number of events (clocks) on the PA[0] or PA[3] input are counted. In this mode the counting will start directly on the next active clock edge as selected on the PortA input configuration. Internally the active clock edge is always the positive edge.
The Event Count Mode is switched on by setting bit **EvCount** in the register **RegCCntl2** to '1'.
PA[3] and PA[0] inputs can be inverted depending on register **OPTIntEdgPA** but should be debounced. The debouncer is switched on in register **OPTDebIntPA** bits NoDebIntPA[3]=0 and NoDebIntPA[0]=0 its frequency depends on the bit **DebSel** from register **RegPresc** setting. The inversion of the internal clock signal derived from PA[3] or PA[0] is active with **IntEdgPA[3]** respectively **IntEdgPA[0]** equal to 1.

Figure 19. Internal clock synchronization



7.4 Compare Function

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 a interrupt (**IRQComp**) is generated. The compare function is switched on with the bit **EnComp** in the register **RegCCntl2**. With EnComp = 0 never an IRQComp is generated. Starting the counter with the same value as the compare register is possible, no IRQ is generated. The compare value must be different from hex 0 in up-count mode and different from hex 3FF (resp. FF, 3F, F if limited bit counting) in down-count mode. Full or Limited bit compare are possible, defined by bit **SelIntComp** in register **RegSysCntl1**. The bit **EnComp** is reset with every load operation (**Load** = 1).

Full bit compare function.
To be in this mode, set the bit **SelIntComp** 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. Be careful, **PWMOn** also redefines the PortB PB[3] output data. (refer to section 7.5).

Limited bit compare
This is the default, with the bit **SelIntComp** set to '0' the compare function will only take as many bits into account as defined by the counter length (**BitSel[1:0]**) selection (see chapter 7.1).

7.5 Pulse Width Modulation (PWM) Generation

The PWM generator uses the behavior of the Compare function so **EnComp** must be set to activate the PWM function. At each Roll Over or Compare Match the PWM state - which is output on PortB PB[3] - will toggle. The start value on PB[3] is depending on the **Up/DownCount** Mode.
Setting **PWMOn** to '1' in register **RegPresc** routes the counter PWM output to PortB PB[3]. Insure that PB[3] is set to Output mode (refer to section 6.4 for the PortB setup). After using the PWM function, do not forget to reset the bit **PWMOn** in order to have PB[3] in a normal mode.
The PWM signal generation is independent of the limited or full bit compare selection bit **SelIntComp.** However if SelIntComp=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 with the unused counter bits. It will count from the start value until the **IRQComp** match.

For instance, loading the counter in UpCount 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, (BitSel[1,0])
- bits [9:4] are the unused counter bits = 05, (nbr 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 SelIntComp=0 (limited bit compare) will produce an unlimited nbr of 2 cycles PWM pulses.

7.5.1 How the PWM generator works.

For UpCount Mode; Setting the counter in UpCount and PWM mode, the PB[3] PWM output is defined to be 0. 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 **SelIntComp** setting which is valid only for the IRQ generation.
In above example the PWM starts with '0' (UpCount),
2 cycles later Compare Match -> PWM to '0',
14 cycles later RollOver -> 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 20. PWM Output in UpCount Mode

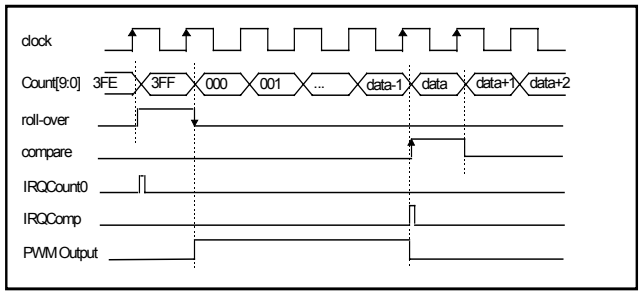
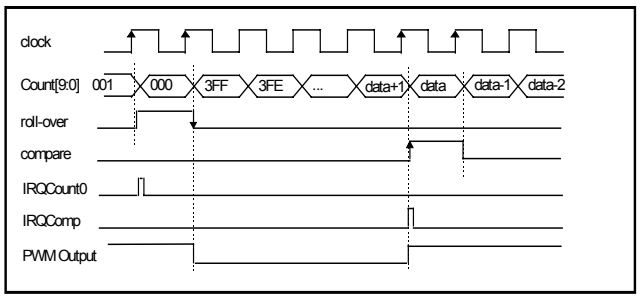


Figure 21. PWM Output in DownCount Mode



In DownCount 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'.

7.5.2 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[19]	-> 53 µs	(600kHz)
the maximum signal period is	: 1024 x Fmin*	-> 1024 x 1/ck[1]	-> 1024 s	(600kHz)
the minimal pulse width is	: 1 bit	-> 1 x 1/ck[19]	-> 3.3µs	(600kHz)

* This values are for Fmax or Fmin derived from the internal system clock (600kHz). Much shorter (and longer) PWM pulses can be achieved by using the PortA as frequency input (undebounced input).

7.6 Counter setup

RegCDataL[3:0], RegCDataM[3:0], RegCDataH[1:0] are used to store the initial count value called **CReg[9:0]** which is written into the count register bits **Count[9:0]** with the **Load** command. **Load** is automatically reset thereafter. The counter value **Count[9:0]** can be read out at any time - except when using nondebounced high frequency PortA input - but 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 nondebounced high frequency PortA 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 (resp. 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;

1st, set the counter into stop mode (**Start**=0).
2nd, select the frequency and Up- or Down 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, PWM)
5th, if compare mode desired , then write **RegCDataL, RegCDataM, RegCDataH** (compare value)
6th, set bit **Start** and **EnComp** in **RegCCntl2**

7.7 10-bit Counter Registers

Table 7.7.1 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 7.7.2 Counter input frequency selection with CountFSel[2..0]

CountFSel2	CountFSel1	CountFSel0	clock source selection
0	0	0	Port A PA[0]
0	0	1	Prescaler ck[19]
0	1	0	Prescaler ck[16]
0	1	1	Prescaler ck[13]
1	0	0	Prescaler ck[10]
1	0	1	Prescaler ck[4]
1	1	0	Prescaler ck[1]
1	1	1	Port A PA[3]

Table 7.7.3 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 7.7.4 register **RegSysCntl1**

Bit	Name	Reset	R/W	Description
3	*IntEn*	*0*	*R/W*	*general interrupt enable*
2	*SLEEP*	*0*	*R/W*	*Sleep mode*
1	SelIntComp	0	R/W	compare Interrupt select
0	*ChTmDis*	*0*	*R/W*	*for EM Test only*

Default : interrupt on limited bit compare

Table 7.7.5 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 7.7.6 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 7.7.7 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

8. Serial (Output) Write Buffer - SWB

The EM6640 has Serial Write Buffer which outputs serial data and serial clock coming from prescaler or a other mode where the clock can be fed from outside (PA[0]).

The SWB is enabled by setting **EnSWB** bit in **RegSWBCntl** and by setting the PortC in output mode.

Serial Write Buffer clock is selected by the **SWBFSel0** and **SWBFSel1** bits in the **RegSWBCntl** register.

TestVar[3] signal, which is used to make conditional jumps, indicates "Transmission finished" in automatic send mode or "SWBbuffer empty " interrupt in interactive send mode. In interactive mode, TestVar[3] is equivalent to the interrupt request flags stored in **RegIRQi** registers : it permits to recognize the interrupt source. (See also the interrupt handling section 10 for further information). To serve the "SWBbuffer empty " interrupt request, one only has to make a conditional jump on TestVar[3].

On the receiver side, serial data can be evaluated on serial clock falling edge. Internally, new data goes out on rising edge of the same clock.

By default the SWBdata level (on PPC[1]) will be equal to VDD as are all the others terminals. Depending on the metal1 option **MSWBdataLevel** (refer to SWBdataLevel Option, on page 50), the level of PPC[1] can be selected as VregLogic. When this metal option is in place, VregLogic is increased by ~50mV.
In this case, a high current can be drawn from this terminal without penalizing the output level (see DC characteristics on page 57).
This allows to transmit the data through a RF module for instance, without an additional regulator or amplifier.

To guarantee the electrical characteristics when PPC[1] is selected as VregLogic, VDD should not be below 2.2v to output 2mA. If the PPC[1] terminal is used as an input, it's input high level has to be VregLogic.

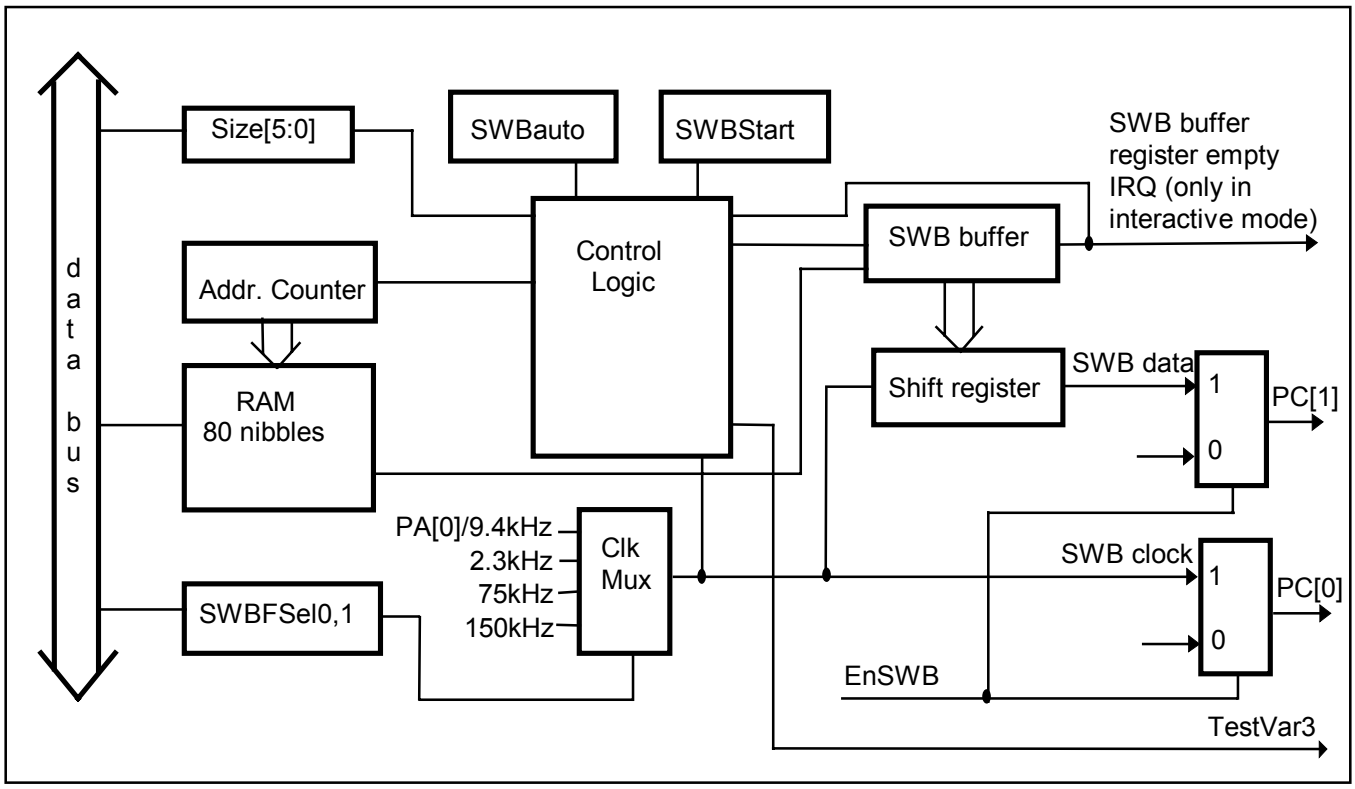


Figure 22 Serial write buffer

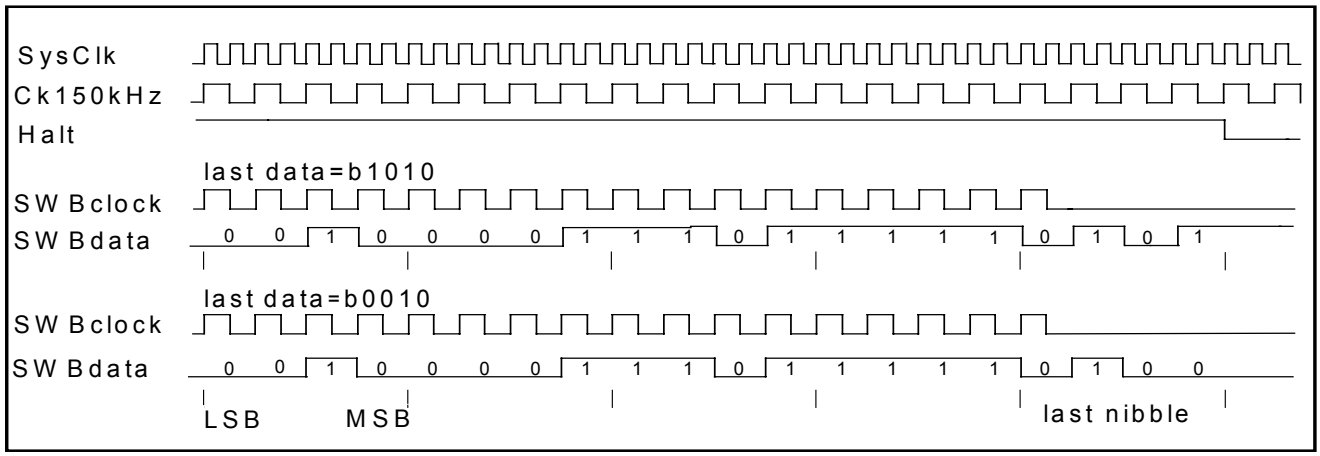
The Serial Write Buffer has two possible working modes :

8.1 SWB Automatic send mode

In this mode, one has first to prepare data to be sent (up to 256 bits), select the clock and write the bit **SWBAuto** in **RegSWBSizeH** register. Then, executing HALT instruction starts the sending operation. During automatic sending, one can not do anything with the microcontroller (instructions can not be executed because of STANDBY mode). At the end of transmission, EM6640 comes out of STANDBY mode and generates a high level on TestVar[3] meaning “transmission finished“.
This mode has additional features compared to interactive mode. In order to be fully flexible, one can decide to send for the last package a number of clocks not egal to 4. The number goes from 1 to 4 clocks depending on bit **DiscCk[1:0]** in register **RegSWBCntl2**. All four data bits are always transmitted independent of the number of clocks selected.
The level of the last bit of the last package is automatically latched as long as the SWB is enabled.

Figure 23 represents two ends of transmission in automatic mode: 1st with data=b1010 and 2nd with data=0010 for the last nibble. Bit **DiscCk[0]** and bit **DiscCk[1]** are set to ‘1’, so only one SWBclock will be send for the last package.

Figure 23 exemple of 2 ends of transmission (SysClk is shown two times longer for a better visibility)



8.1.1 SWB Automatic with external clock

In automatic mode, setting **SWBFSel0** and **SWBFSel1** to ‘0’ in register **RegSWBCntl**, selects an external clock from the PA[0] input which can be used as a serial transmission clock. Serial clock and serial data will go out at the same frequency as the clock provided on the PA[0] input. Depending of the value of **CkExtDiv[1:0]** bits in register **RegSWBCntl**, the external clock applied on PA[0] can be divided internally by 1 (default), 4, 88 or 352 times.
The PA[0] input can be inverted depending on the register **OPTIntEdgPA** and debounced depending on the register **OPTDebIntPA**. Refer to chapter IRQ on portA, page 16.
For **external clock/1** selection, the transmission will start on the third positive clock edge or on the second negative clock edge applied on PA[0], following the HALT command. This is depending of the PA[0] input configuration, positive edge for the first case and negative edge for the second case.
For **external clock/4**, **external clock/88** and **external clock/352** selection, the transmission will start on the half clock edge selection + 2 clocks (resp. 4th external clock, 46th external clock and 178 external clock) following the HALT command.

8.1.2 How the SWB in automatic mode works

Data to be sent must be prepared in the following order:

1st: First 4-bit package must be written in **RegSWBuff** register.

2nd: Other 4-bit packages must be loaded in the RAM from address 0 ==> second package at address 0, third at address 1... (the maximum address space for SWB is 3E hex ==> 64 4-bit packages = 256 bits).

3rd: One has to load the data size (address of the last 4-bit package) into **RegSWBSizeL** and **RegSWBSizeH** registers and has to set to "1" at the same time the bit **SWBAuto**. Therefore, the number of 4-bit packages which can be sent ranges from 2 (size = 0) to 64 (size = 3E hex).

4th: To start the transmission, one has to put the EM6640 in HALT mode.

At the end of transmission, write any data to **RegSWBuff** in order to clear TESTvar[3] which allows the SWB to be written again.

By writing **SWBAuto** to "1", general interrupt enable flag **IntEn** is disabled until the end of SWB transmission and therefore no interrupt can occur before transmission ends. Consequently, a HALT command should follow a **SWBAuto** write in order not to miss any interruptions. When the transmission is finished, the bit **SWBAuto** is cleared and TESTvar[3] goes high and stays at this level until **RegSWBuff** register is written.

SysClk is shown two times longer for a better visibility !

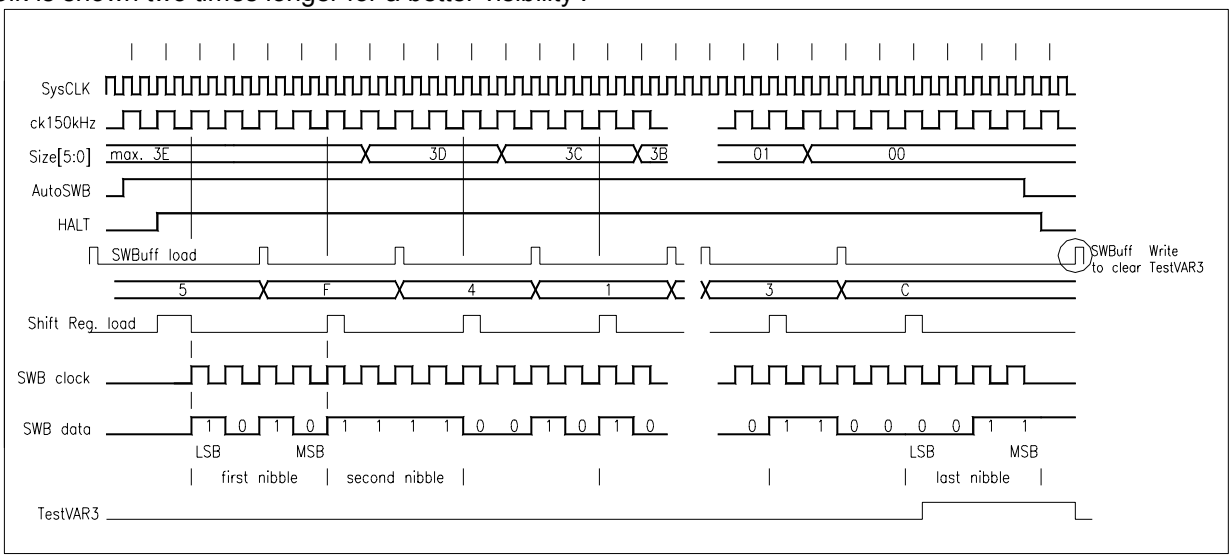


Figure 24 Automatic Serial Write Buffer transmission

At the end of transmission the general interrupt enable flag **IntEn** is set to its active high state again when HALT becomes inactive. An internal START signal enables the CPU to come out of HALT and execute the following instruction. However if an interrupt occurs during SWB transmission, the CPU will service this first.

One can restart the transmission with the same data by simply reloading the register **RegSWBuff** and then the registers **RegSWBSizeL** and **RegSWBSizeH** (with bit **SWBauto** set to "1"), disabling the general interrupt enable flag and putting again the EM6640 into HALT mode.

8.2 SWB Interactive send mode

In interactive send mode, data to be sent must be loaded in **RegSWBuff** register (no RAM space is used for serial transmission).

8.2.1 How the SWB in interactive mode works

One has first to select the serial transmission clock in **RegSWBCntl** register and load the first 4-bit package to be sent in **RegSWBuff** register.
Then setting to “1“ the bit **SWBStart** in **RegSWBSizeH** register starts the transmission : data are transferred from **RegSWBuff** register to shift register and put on serial data output. When **RegSWBuff** register is empty, an interrupt request **SWBempty** which *can not be masked* is generated and TestVar[3] goes high.

According to the selected clock, one has to take into account how many instructions are needed to process **SWBempty** interrupt request : recognize this interrupt (conditional jump on TESTvar[3]) and load the new 4-bit package in **RegSWBuff** register.
Indeed, If processing of SWBempty interrupt request is too long and internal SHIFT register is empty before new 4-bit package was written in **RegSWBuff** register, the transmission is broken : serial clock output stops at "0", bit **SWBStart** and SWBempty are cleared to “0“ and TestVar[3] remains to “1“. This could be the case for a 150kHz transmission speed. The application must restart the serial transmission by writing the **SWBStart** in **RegSWBSizeH** register after writing the next nibble to the **RegSWBuff** register

SWBempty and TestVar[3] are cleared to "0" at each **RegSWBuff** register writing operation.
After loading the last nibble in the **RegSWBuff** register a new interrupt is generated when this data is transferred to an intermediate Shift Register. Precaution must be made in this case because the SWB will give repetitive interrupts until the last data is sent out completely and the **SWBStart** bit goes low automatically. One possibility to overcome this is to check in the Interrupt subroutine that the **SWBStart** bit went low before exiting interrupt.

At the end of transmission, one must write **RegSWBuff** register to set to “0“ TESTvar[3].

SysClk is shown two times longer for a better visibility !

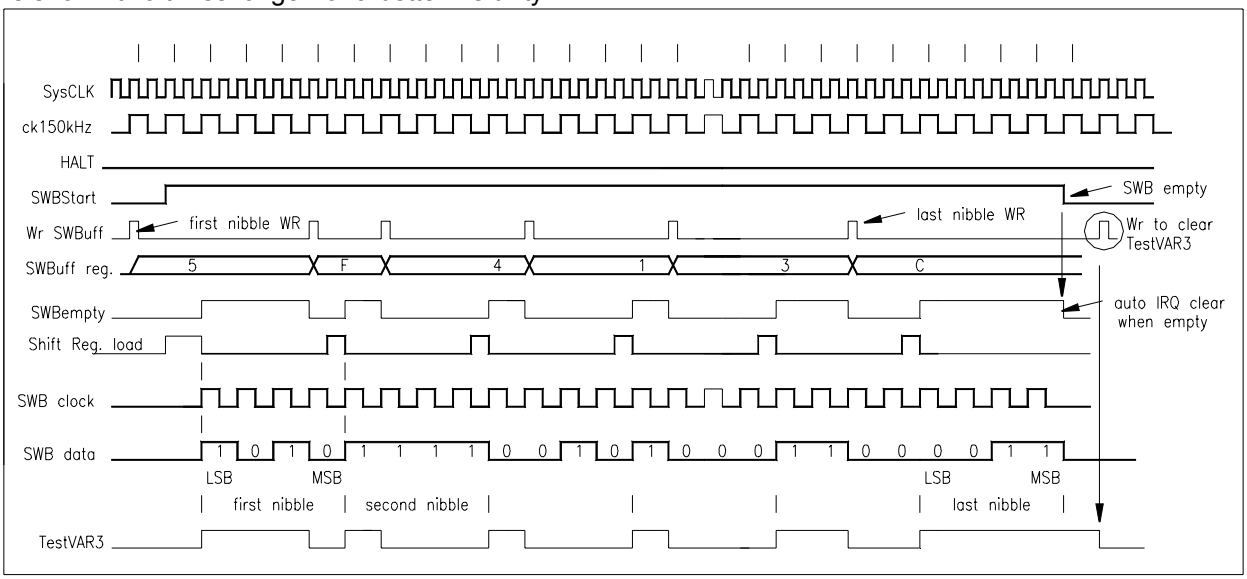


Figure 25 Interactive Serial Write Buffer mode

N.B. In interactive mode, the number of clocks per package is always equal to 4 and the level of the last bit of the last package is not latched. These 2 features are valid only in automatic mode.

8.3 SWB registers

Table 8.3.1 SWB clock selection register **RegSWBCntl**

Bit	Name	Reset	R/W	Description
3	EnSWB	0	R/W	enable SWB
2	--			
1	SWBFSel1	0	R/W	SWB clock selection
0	SWBFSel0	0	R/W	SWB clock selection

Table 8.3.2 Serial Write Buffer clock selection

SWB clock output	SWBFSel1	SWBFSel0
9375 Hz/CkExt	0	0
2344 Hz	0	1
75000 Hz	1	0
150000 Hz	1	1

For the first selection, 9375 Hz is dedicated to interactive mode and CkExt to automatic mode.

Table 8.3.3 SWB external clock and discard clock selection in register **RegSWBCntl2**

Bit	Name	Reset	R/W	Description
3	CkExtDiv[1]	0	R/W	SWB external clock divider selection
2	CkExtDiv[0]	0	R/W	SWB external clock divider selection
1	DiscCk[1]	0	R/W	SWB discard clock selection
0	DiscCk[0]	0	R/W	SWB discard clock selection

Table 8.3.4 Serial Write Buffer discard clock selection

Number of clocks during the last nibble	DiscCk[1]	DiscCk[0]
4 clocks	0	0
3 clocks	0	1
2 clocks	1	0
1clock	1	1

Table 8.3.5 Serial Write Buffer external clock divider selection

SWB clock output	CkExtDiv[1]	CkExtDiv[0]
external clock / 1	0	0
external clock / 4	0	1
external clock / 88	1	0
external clock / 352	1	1

Table 8.3.6 SWB buffer register **RegSWBuff**

Bit	Name	Reset	R/W	Description
3	Buff[3]	1	R/W	SWB buffer bit 3
2	Buff[2]	1	R/W	SWB buffer bit 2
1	Buff[1]	1	R/W	SWB buffer bit 1
0	Buff[0]	1	R/W	SWB buffer bit 0

Table 8.3.7 SWB Low size register **RegSWBSizeL**

Bit	Name	Reset	R/W	Description
3	Size[3]	0	R/W	auto mode buffer size bit3
2	Size[2]	0	R/W	auto mode buffer size bit2
1	Size[1]	0	R/W	auto mode buffer size bit1
0	Size[0]	0	R/W	auto mode buffer size bit0

Table 8.3.8 SWB High size register **RegSWBSizeH**

Bit	Name	Reset	R/W	Description
3	SWBAuto	0	R/W	SWB Automatic mode select
2	SWBStart	0	R/W	SWB Start interactive mode
1	Size[5]	0	R/W	auto mode buffer size bit5
0	Size[4]	0	R/W	auto mode buffer size bit4

9. EEPROM

The EM6640's EEPROM contains 32 words of 8 bits each. Addressing is done indirectly using 5 bits (32 addresses) defined in **RegEEPAddr** and **RegEEPCntl** registers.
Either in erase/write mode or in read mode, the EEPROM will be functional for all the standard operating conditions. Refer to EM6640 Electrical specifications, on page 55.
In **RegEEPCntl** register, one can select EEPROM reading or writing operation by setting respectively to "0" or "1" the bit **EEPRdWr**.

How to read data from EEPROM :
1st instr. : write EEPROM address (4 low bits) in **RegEEPAddr** register.
2nd instr. : write the high address bit and select reading operation in **RegEEPCntl** register.
3rd instr. : read EEPROM low data in **RegEEPDataL** register.
4th instr. : read EEPROM high data in **RegEEPDataH** register.
The two last instructions can be executed in the reverse order.

How to write data in EEPROM :
1st instr. : write EEPROM address (4 low bits) in **RegEEPAddr** register.
2nd instr. : write EEPROM low data in **RegEEPDataL** register.
3rd instr. : write EEPROM high data in **RegEEPDataH** register.
4th instr. : write the high address bit and select writing operation in **RegEEPCntl** register.
The three first instructions can be executed in any order.

Writing to the **RegEEPCntl** register automatically starts the EEPROM reading or writing operation according to the bit **EEPRdWr**. To guarantee correct functionality when the EEPROM is being used, one should never modify the EEPROM's registers.

EEPROM reading operation lasts 10µs : then, data are available in **RegEEPDataL** and **RegEEPDataH** registers. The flag **EEPRdBusy** in **RegEEPCntl** register stays high until the reading operation is finished.

EEPROM writing operation lasts 20200µs : 10000µs for erase process, 10000µs for effective write process and 200µs between these two processes. The flag **EEPWrBusy** in **RegEEPCntl** register stays high until the writing operation is finished. An interrupt request **IRQEEP** is generated at the end of each writing operation. This interrupt request can be masked (default, **MaskIRQEEP** bit). See also the interrupt handling section 10 for further information.

During reading operation, the device will drawn an additional 60µA of Ivdd current and during erasing/writing operation an additional 45µA of Ivdd current (typical, @ 3V, 600kHz, 25°C).

N.B. : During a EEPROM writing operation, all the peripherals of the EM6640 can be used but one should never put the circuit in SLEEP mode or execute a RESET.

9.1 EEPROM registers

Table 9.1.1 EEPROM control register **RegEEPCntl**

Bit	Name	Reset	R/W	Description
3	EEPRdBusy	0	R	EEPROM reading operation busy flag
2	EEPWrBusy	0	R	EEPROM writing operation busy flag
1	EEPRdWr	0	R/W	EEPROM operation read=0 / write=1
0	Addr[4]	0	R/W	EEPROM address bit 4

Writing this register starts automatically EEPROM reading or writing operation

Table 9.1.2 EEPROM address register **RegEEPAddr**

Bit	Name	Reset	R/W	Description
3	Addr[3]	0	R/W	EEPROM address bit 3
2	Addr[2]	0	R/W	EEPROM address bit 2
1	Addr[1]	0	R/W	EEPROM address bit 1
0	Addr[0]	0	R/W	EEPROM address bit 0

Table 9.1.3 EEPROM data low register **RegEEPDataL**

Bit	Name	Reset	R/W	Description
3	EEPdata[3]	0	R/W	EEPROM data bit 3
2	EEPdata[2]	0	R/W	EEPROM data bit 2
1	EEPdata[1]	0	R/W	EEPROM data bit 1
0	EEPdata[0]	0	R/W	EEPROM data bit 0

Table 9.1.4 EEPROM data high register **RegEEPDataH**

Bit	Name	Reset	R/W	Description
3	EEPdata[7]	0	R/W	EEPROM data bit 7
2	EEPdata[6]	0	R/W	EEPROM data bit 6
1	EEPdata[5]	0	R/W	EEPROM data bit 5
0	EEPdata[4]	0	R/W	EEPROM data bit 4

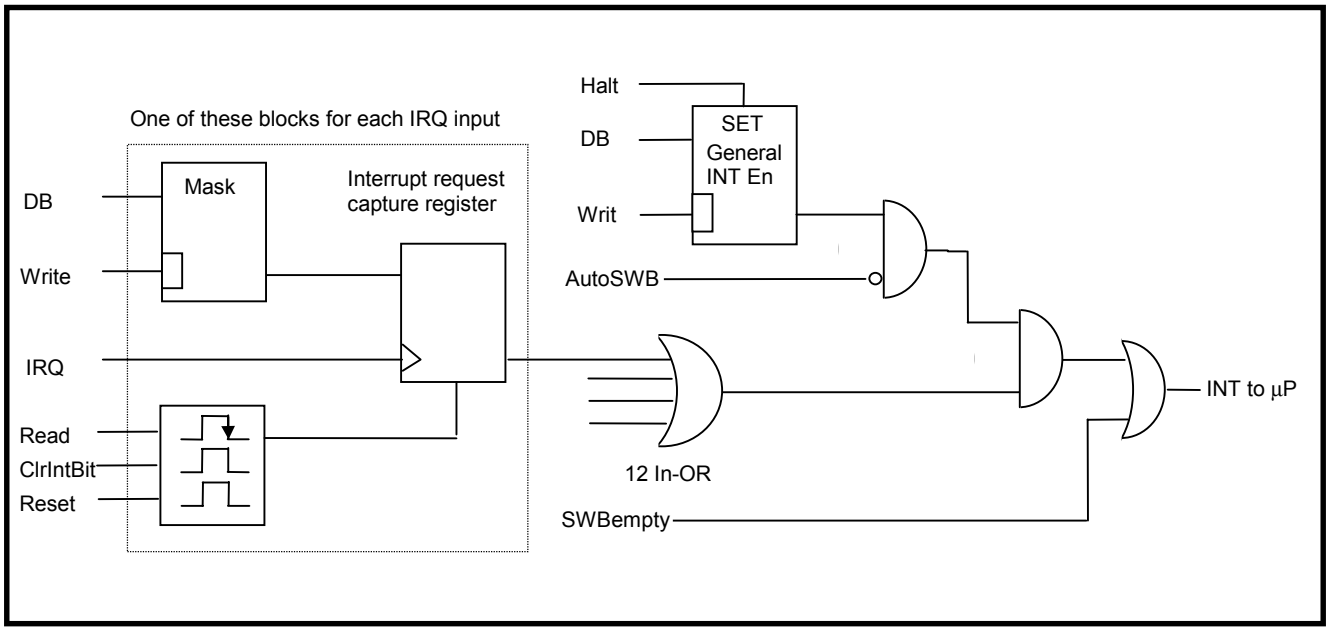
10. Interrupt Controller

The EM6640 has 12 different interrupt request sources individually maskable. These are :

External(4)	**- PortA(4)**	**PA[3] .. PA[0] inputs**
Internal(8)	**- Prescaler(3)**	**9.4kHz, 586Hz, 1Hz**
	- 10-bit Counter(2)	**Count0, CountComp**
	-SWB(1)	**SWBuff empty in interactive mode**
	-SVLD(1)	**End of measure**
	-EEPROM(1)	**End of writing operation**

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 26. Interrupt Controller Block Diagram



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**, **RegIRQ2** or **RegIRQ3** 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 come 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 EM6640 goes into HALT Mode the **IntEn** bit is automatically set to 1, thus allowing to resume from Halt Mode with an interrupt.

10.1 Interrupt control registers

Table 10.1.1 register **RegIRQ1**

Bit	Name	Reset	R/W	Description
3	IRQPA[3]	0	R/W*	PortA PA[3] interrupt request
2	IRQPA[2]	0	R/W*	PortA PA[2] interrupt request
1	IRQPA[1]	0	R/W*	PortA PA[1] interrupt request
0	IRQPA[0]	0	R/W*	PortA PA[0] interrupt request

W* ; Writing of 1 clears the corresponding bit.

Table 10.1.2 register **RegIRQ2**

Bit	Name	Reset	R/W	Description
3	IRQ1Hz	0	R/W*	Prescaler interrupt request
2	IRQ586Hz	0	R/W*	Prescaler interrupt request
1	IRQ9k4Hz	0	R/W*	Prescaler interrupt request
0	IRQEE	0	R/W*	EEPROM interrupt request

W* ; Writing of 1 clears the corresponding bit.

Table 10.1.3 register **RegIRQ3**

Bit	Name	Reset	R/W	Description
3	---			
2	IRQVLD	0	R/W*	SVLD interrupt request
1	IRQCount0	0	R/W*	Counter interrupt request
0	IRQCntComp	0	R/W*	Counter interrupt request

W* ; Writing of 1 clears the corresponding bit.

Table 10.1.4 register **RegIRQMask1**

Bit	Name	Reset	R/W	Description
3	MaskIRQPA[3]	0	R/W	PortA PA[3] interrupt mask
2	MaskIRQPA[2]	0	R/W	PortA PA[2] interrupt mask
1	MaskIRQPA[1]	0	R/W	PortA PA[1] interrupt mask
0	MaskIRQPA[0]	0	R/W	PortA PA[0] interrupt mask

Interrupt is not stored if the mask bit is 0.

Table 10.1.5 register **RegIRQMask2**

Bit	Name	Reset	R/W	Description
3	IRQ1Hz	0	R/W	Prescaler interrupt mask
2	IRQ586Hz	0	R/W	Prescaler interrupt mask
1	IRQ9k4Hz	0	R/W	Prescaler interrupt mask
0	IRQEE	0	R/W	EEPROM interrupt request

Interrupt is not stored if the mask bit is 0.

Table 10.1.6 register **RegIRQMask3**

Bit	Name	Reset	R/W	Description
3	---			
2	MaskIRQVLD	0	R/W	SVLD interrupt mask
1	MaskIRQCount0	0	R/W	Counter interrupt mask
0	MaskIRQCntComp	0	R/W	Counter interrupt mask

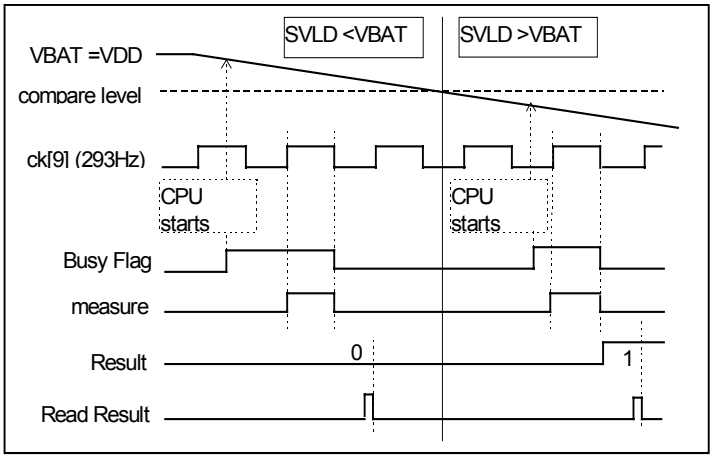
Interrupt is not stored if the mask bit is 0

11. Supply Voltage Level Detector

The EM6640 has a built-in Supply Voltage Level Detector (SVLD), such that the CPU can compare the supply voltage against a pre-selected value (default is 2.2V or 2.5V, register selectable). The pre-selected values can be adjusted by metal mask options. During Sleep Mode this function is inhibited.

The CPU activates the supply voltage level detector by writing **VldStart**=1 in the register **RegVldCntl**. The actual measurement starts on the next ck[9] rising edge and lasts during the ck[9] high period (1.7ms at 600kHz). The busy flag **VldBusy** stays high from **Start** set until the measurement is finished. The worst case time until the result is available is 1.5 x ck[9] prescaler clock periods (600kHz -> 5.1ms). The detection level must be defined in register **RegVldCntl** before the **Start** bit is set.
During the actual measurement the device will draw an additional 5uA of I_{VDD} current. After the end of the measure an interrupt request **IRQVLD** is generated and the result is available by inspection of the bit **VLDResult**. If the result is read 0, then the power supply voltage was greater than the detection level value. If read 1, the power supply voltage was lower than the detection level value. During each read while **Busy=1** the **VLDResult** is not defined.

Figure 27. SVLD Timing Diagram



11.1 Supply Voltage Level Detector Register

Table 11.1.1 register **RegVldCntl**

Bit	Name	Reset	R/W	Description
3	VLDResult	0	R*	VLD result flag
2	VLDStart	0	W	VLD start
2	VLDBusy	0	R	VLD busy flag
1	*0*	*0*	*R*	
0	VLDlevel	0	R/W	VLD level selection

R* ; VLDResult is not guaranteed while VLDBusy=1

Table 11.1.2 Voltage detector value selecting

VLDlevel	typical voltage level
0	2.5V
1	2.2V

12. RAM

The EM6640 has one 80x4 bit RAM built-in located on addresses hex 0 to 4F. All the RAM nibbles are direct addressable.

Figure 28. RAM Architecture

Address RAM	Bits Read/Write			
79	3	2	1	0
78	3	2	1	0
77	3	2	1	0
76	3	2	1	0
•				
•				
•				
03	3	2	1	0
02	3	2	1	0
01	3	2	1	0
00	3	2	1	0

80x4 directly addressable RAM

12.1 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.

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		
Ram	00 . . . 4F	00 . . . 79	xxxx	0 : data0 1 : data1 2 : data2 3 : data3		direct addressable Ram 80x4
RegEEPDataL	50	80	0000	0 : EEPData[0] 1 : EEPDatal[1] 2 : EEPData[2] 3 : EEPDatal[3]		EEPROM data low bits
RegEEPDataH	51	81	0000	0 : EEPData[4] 1 : EEPDatal[5] 2 : EEPData[6] 3 : EEPDatal[7]		EEPROM data high bits
RegEEPAddr	52	82	0000	0 : Addr[0] 1 : Addr[1] 2 : Addr[2] 3 : Addr[3]		EEPROM address 4 first bits
RegEEPCntl	53	83	0000	0 : Addr[4] 1 : EEPRdWr 2 : EEPWrBusy 3 : EEPRdBusy	0 : Addr[4] 1 : EEPRdWr 2 : -- 3 : --	EEPROM address 5th bit. EEPROM control : Read/Write EEPROM Write busy flag EEPROM Read busy flag
RegPA	54	84	xxxx	0 : PAData[0] 1 : PAData[1] 2 : PAData[2] 3 : PAData[3]	----	read PortA directly
RegPBCntl	55	85	0000	0 : PBIOCntl[0] 1 : PBIOCntl[1] 2 : PBIOCntl[2] 3 : PBIOCntl[3]		PortB Control default : input mode
RegPBData	56	86	0000	0 : PB[0] 1 : PB[1] 2 : PB[2] 3 : PB[3]	0 : PBData[0] 1 : PBData[1] 2 : PBData[2] 3 : PBData[3]	PortB data output. Pin PortB read default : 0
RegPCCntl	57	87	0000	0 : PCIOCntl[0] 1 : PCIOCntl[1] 2 : PCIOCntl[2] 3 : PCIOCntl[3]		PortC Control default : input mode
RegPCData	58	88	0000	0 : PC[0] 1 : PC[1] 2 : PC[2] 3 : PC[3]	0 : PCData[0] 1 : PCData[1] 2 : PCData[2] 3 : PCData[3]	PortC data output. Pin PortC read default : 0
RegSWBCntl	59	89	0000	0 : SWBFSel0 1 : SWBFSel1 2 : '0' 3 : EnSWB	0 : SWBFSel0 1 : SWBFSel1 2 : -- 3 : EnSWB	SWB control : clock selection, enable SWB

Register name	add hex	add dec	reset value	read_bits	write_bits	Remarks
			b'3210	Read/Write_bits		
RegSWBuff	5A	90	1111	0 : Buff[0] 1 : Buff[1] 2 : Buff[2] 3 : Buff[3]		SWB buffer register
RegSWBSizeL	5B	91	0000	0 : Size[0] 1 : Size[1] 2 : Size[2] 3 : Size[3]		SWB size low bits
RegSWBSizeH	5C	92	0000	0 : Size[4] 1 : Size(5] 2 : SWBStart 3 : SWBAuto		SWB size high bits. Automatic/interactive mode selection.
RegCCntl1	5D	93	0000	0 : CountFSel0 1 : CountFSel1 2 : CountFSel2 3 : UP/Down		10 bit counter control 1 ; frequency and up/down
RegCCntl2	5E	94	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	5F	95	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 bits
RegCDataM	60	96	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 bits
RegCDataH	61	97	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
RegIRQMask1	62	98	0000	0 : MaskIRQPA[0] 1 : MaskIRQPA[1] 2 : MaskIRQPA[2] 3 : MaskIRQPA[3]		PortA interrupt mask ; masking active low
RegIRQMask2	63	99	0000	0 : MaskIRQEE 1 : MaskIRQ9k4Hz 2 : MaskIRQ586Hz 3 : MaskIRQ1Hz		prescaler interrupt mask ; masking active low
RegIRQMask3	64	100	0000	0 : MaskIRQCntComp 1 : MaskIRQCount0 2 : MaskIRQVLD 3 : '0'	0 :MaskIRQCntComp 1 : MaskIRQCount0 2 : MaskIRQVLD 3 : --	10 bit counter, VLD interrupt mask masking active low
RegIRQ1	65	101	0000	0 : IRQPA[0] 1 : IRQPA[1] 2 : IRQPA[2] 3 :IRQPA[3]	0 : RIRQPA[0] 1 : RIRQPA[1] 2 : RIRQPA[2] 3 : RIRQPA[3]	Read : PortA interrupt Write : Reset IRQ if data bit = 1.
REgIRQ2	66	102	0000	0 : IRQEE 1 : IRQ9k4Hz 2 : IRQ586Hz 3 : IRQ1Hz	0 : RIRQEE 1 : RIRQ9k4Hz 2 : RIRQ586Hz 3 : RIRQ1Hz	Read : prescaler IRQ , EEPROM IRQ Write : Reset IRQ if data bit = 1
RegIRQ3	67	103	0000	0 :IRQCntComp 1 : IRQCount0 2 : IRQVLD 3 : '0'	0 : RIRQCntComp 1 : RIRQCount0 2 : RIRQVLD 3 : --	Read : 10 bit counter, VLDl interrupt Write : Reset IRQ if data bit =1.

Register name	add hex	add dec	reset value	read_bits	write_bits	Remarks
			b'3210	Read/Write_bits		
RegSysCntl1	68	104	00x0	*0 : ChTmDis* 1 : SelintComp 2 : '0' 3 : IntEn	*0 : ChTmDis* 1 : SelintComp 2 : Sleep 3 : IntEn	system control 1 *ChTmDis only usable for EM test modes with Test=1*
RegSysCntl2	69	105	0000	0 : WDVal0 1 : WDVal1 2 : SleepEn 3 : '0'	0 : -- 1 : -- 2 : SleepEn 3 : WDReset	system control 2 ; watchdog value and periodical reset, enable sleep mode
RegSysCntl3	6A	106	0000	0 : NoLogicWD 1 : NoOscWD 2 : '0' 3 : '0'	0 : NoLogicWD 1 : NoOscWD 2 : -- 3 : --	system control 3 ; watchdog control
RegPresc	6B	107	0000	0 : DebSel 1 : '0' 2 : '0' 3 : PWMOn	0 : DebSel 1 : -- 2 : ResPresc 3 : PWMOn	prescaler control ; debouncer and prescaler interrupt selection
RegVLDCntl	6C	108	0000	0 : VLDlevel 1 : '0' 2 : VLDBusy 3 : VLDResult	0 : VLDlevel 1 : -- 2 : VLDStart 3 : --	VLD control : level detection, start (busy flag) and result.
RegSWBCntl2	6D	109	0000	0 : DiscCk[0] 1 : DiscCk[1] 2 : CkExtDiv[0] 3 : CkExtDiv[1]		SWB control2 : discard clock selection external clock divider selection
IXLow	6E	110	xxxx	0 : IXLow[0] 1 : IXLow[1] 2 : IXLow[2] 3 : IXLow[3]		internal µP index register low nibble ;
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 ;

14. Option Register Memory Map

The values of the Option Registers are set only by initial reset on power up and through write operations. Other resets as reset from watchdog, reset from input PortA, etc... do not change the Options Register value.

Register name	add hex	add dec	power up value	read_bits	write_bits	Remarks
			b'3210	Read/Write_bits		
OPTSlpCntRst OPT[47 :44]	72	114	0000	0 : RstSlpTSel[0] 1 : RstSlpTSel[1] 2 : RstSlpTSel[2] 3 : EnSlpCntRst[3]		option register ; Sleep Counter Reset default : SCR disable
OPTDebIntPA OPT[3 :0]	73	115	0000	0 : NoDebIntPA[0] 1 : NoDebIntPA[1] 2 : NoDebIntPA[2] 3 : NoDebIntPA[3]		option register ; debouncer on PortA for interrupt gen. Default : debouncer on
OPTIntEdgPA OPT[7 :4]	74	116	0000	0 : IntEdgPA[0] 1 : IntEdgPA[1] 2 : IntEdgPA[2] 3 : IntEdgPA[3]		option register ; interrupt edge select on PortA default : pos edge
OPTNoPullPA OPT[11 :8]	75	117	0000	0 : NoPdPA[0] 1 : NoPdPA[1] 2 : NoPdPA[2] 3 : NoPdPA[3]		option register ; pull-up/down selection on PortA default : pull-down
OPTNoPdPB OPT[15 :12]	76	118	0000	0 : NoPdPB[0] 1 : NoPdPB[1] 2 : NoPdPB[2] 3 : NoPdPB[3]		option register ; pulldown selection on PortB default : pull-down
OPTNoPdPC OPT[19 :16]	77	119	0000	0 : NoPdPC0] 1 : NoPdPC[1] 2 : NoPdPC[2] 3 : NoPdPC[3]		option register ; pulldown selection on PortC default : pull-down
OPTNchOpDPB OPT[23 :20]	78	120	0000	0 : NchOpDPB[0] 1 : NchOpDPB[1] 2 : NchOpDPB[2] 3 : NchOpDPB[3]		option register ; n-channel open drain output on PortB default : CMOS output
OPTNchOpDPC OPT[27 :24]	79	121	0000	0 : NchOpDPC[0] 1 : NchOpDPC[1] 2 : NchOpDPC[2] 3 : NchOpDPC[3]		option register ; n-channel open drain output on PortC default : CMOS output
OPTPaRST OPT[31 :28]	7A	122	0000	0 : NoInputReset 1 : SelinpResMod 2 : OscAdj[4] 3 : OscAdj[5]		-PortA input reset option, input reset mode (Or or AND logic). -Adjustment of RC oscillator (2 MSB) in EEPROM
OPTOscAdj OPT[35 :32]	7B	123	0000	0 : OscAdj[0] 1 : OscAdj[1] 2 : OscAdj[2] 3 : OscAdj[3]		option register ; Adjustment of RC oscillator (4 LSB) in EEPROM
OPTFselPB OPT[39 :36]	7C	124	0000	0 : PB600kHzOut 1 : PB37k5HzOut 2 : PB2k3HzOut 3 : --		option register ; frequency output on PortB
OPTInpRSel1	7D	125	0000	0 : InpRes1PA[0] 1 : InpRes1PA[1] 2 : InpRes1PA[2] 3 : InpRes1PA[3]		option register ; reset through PortA inputs selection, refer to reset part
OPTInpRSel2	7E	126	0000	0 : InpRes2PA[0] 1 : InpRes2PA[1] 2 : InpRes2PA[2] 3 : InpRes2PA[3]		option register ; reset through PortA inputs selection, refer to reset part
RegTestEM	*7F*	*127*	----	----	*accu*	*for EM test only ;* *write accu on PortA Test = 1*

15. Test at EM - Active Supply Current Test

For this purpose, five instructions at the end of the ROM will be added.

```
Testloop:  STI      00H, 0AH
           LDR      1BH
           NXORX
           JPZ      Testloop
           JMP      00H
```

To stay in the testloop, these values must be written in the corresponding addresses before jumping in the loop:

```
1BH:    0101b
32H:    1010b
6EH:    0010b
6FH:    0011b
```

Free space after last instruction: JMP 00H (0000)

Remark: empty space within the program are filled with NOP (FOFF).

16. Mask Options

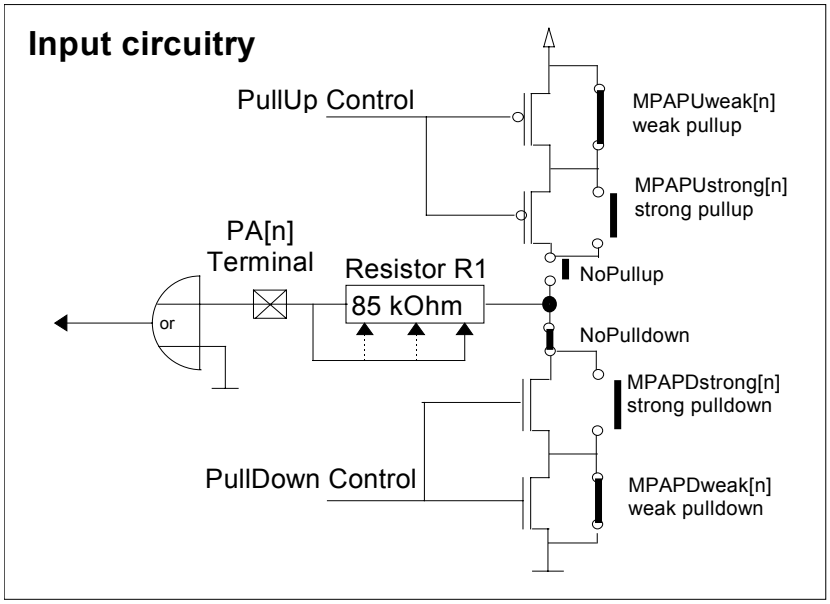
Most options which in many µControllers are realized as metal mask options, are directly user selectable with the option registers allowing a maximum freedom of choice .See chapter : Option Register Memory Map.
The following options can be selected at the time of programming the metal mask ROM.

16.1 Input / Output ports

16.1.1 PortA Metal Options

Pull-Up or No Pull-Up can be selected for each PortA input. A Pull-up selection is excluding a Pull-down on the same input.
Pull-Down or No Pull-Down can be selected for each PortA input. A Pull-down selection is excluding a Pull-up on the same input.
The total pull value (pull-up or pull-down) is a series resistance out of the resistance R1 and the switching transistor. As a switching transistor, the user can choose between a high impedance (weak) or a low impedance (strong) switch (one can not choose both). Weak, strong or none must be chosen. The default is strong. The default resistor R1 value is 85kOhm. The user may choose a different value from 85kOhm down to 0 Ohm. However the value must first be checked and agreed by EM Microelectronic Marin SA.

Figure 29. PortA pull options



Option name		strong Pull-Down	weak Pull-Down	R1 value	NO Pull-Down
		1	**2**	**3**	**4**
MPAPD[3]	PA3 input pull-down				
MPAPD[2]	PA2 input pull-down				
MPAPD[1]	PA1 input pull-down				
MPAPD[0]	PA0 input pull-down				

To select an option put a cross-**X** in column 1,2 and 4 and reconfirm the R1 value in column 3.

The default value is : strong pulldown with R1=85k.
Total value of typ. 100kOhm

Option name		strong Pull-Up	weak Pull-Up	R1 value	NO Pull-Up
		1	**2**	**3**	**4**
MPAPU[3]	PA3 input pull-up				
MPAPU[2]	PA2 input pull-up				
MPAPU[1]	PA1 input pull-up				
MPAPU[0]	PA0 input pull-up				

To select an option put a cross-**X** in column 1,2 and 4 and reconfirm the R1 value in column 3.

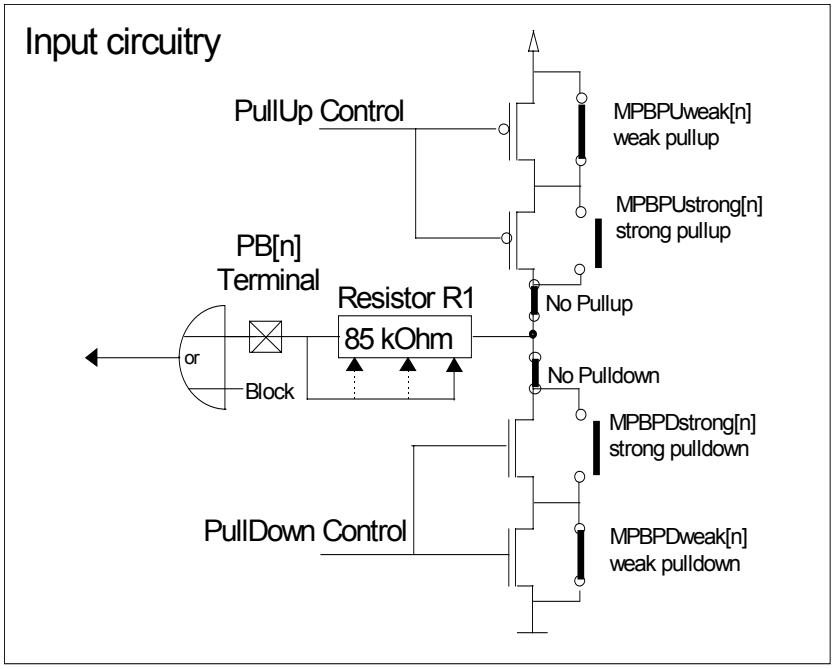
The default value is : strong pull-up with R1=85k
Total value of typ. 100kOhm
By default : no pull-up

16.1.2 PortB Metal Options

Pull-Up or No Pull-Up can be selected for each PortB input. The Pull-Up is only active in N-Channel Open Drain Mode.
Pull-Down or No Pull-Down can be selected for each PortB input.

The total pull value (pull-up or pulldown) is a series resistance out of the resistance R1 and the switching transistor. As a switching transistor the user can choose between a high impedance (weak) or a low impedance (strong) switch (one can not choose both). Weak, strong or none must be chosen. The default is Strong. The default resistor R1 value is 85kOhm. The user may choose a different value from 85kOhm down to 0 Ohm. However the value must first be checked and agreed by EM Microelectronic Marin SA.

Figure 30. PortB pull options



Option name		strong Pull-Down	weak Pull-Down	R1 value	NO Pull-Down
		1	**2**	**3**	**4**
MPBPD[3]	PB3 input pull-down				
MPBPD[2]	PB2 input pull-down				
MPBPD[1]	PB1 input pull-down				
MPBPD[0]	PB0 input pull-down				

To select an option put a cross-**X** in column 1,2 and 4 and reconfirm the R1 value in column 3.

The default value is : strong pulldown with R1=85k
Total value of typ. 100kOhm

Option name		strong Pull-Up	weak Pull-Up	R1 value	NO Pull-Up
		1	**2**	**3**	**4**
MPBPU[3]	PB3 input pull-up				
MPBPU[2]	PB2 input pull-up				
MPBPU[1]	PB1 input pull-up				
MPBPU[0]	PB0 input pull-up				

To select an option put a cross-**X** in column 1,2 and 4 and reconfirm the R1 value in column 3.

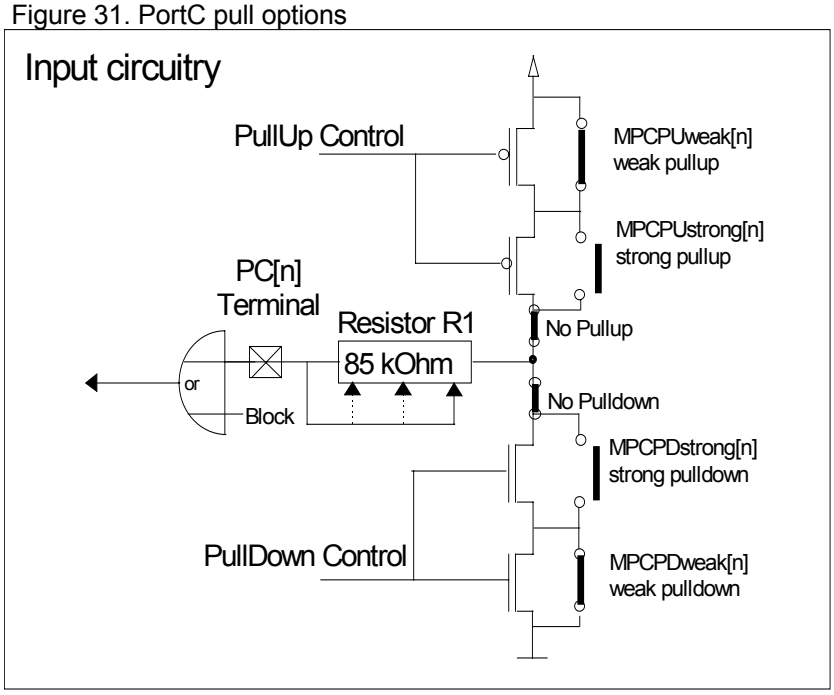
The default value is : strong pull-up with R1=85k
Total value of typ. 100kOhm

16.1.3 PortC Metal Options

Pull-Up or No Pull-Up can be selected for each PortC input. The Pull-Up is only active in N-Channel Open Drain Mode.
Pull-Down or No Pull-Down can be selected for each PortC input.

The total pull value (pull-up or pulldown) is a series resistance out of the resistance R1 and the switching transistor. As a switching transistor the user can choose between a high impedance (weak) or a low impedance (strong) switch (one can not choose both). Weak, strong or none must be chosen. The default is Strong. The default resistor R1 value is 85kOhm. The user may choose a different value from 85kOhm down to 0 Ohm. However the value must first be checked and agreed by EM Microelectronic Marin SA.

Figure 31. PortC pull options



Option name		strong Pull-Down	weak Pull-Down	R1 value	NO Pull-Down
		1	**2**	**3**	**4**
MPBPD[3]	PB3 input pull-down				
MPBPD[2]	PB2 input pull-down				
MPBPD[1]	PB1 input pull-down				
MPBPD[0]	PB0 input pull-down				

To select an option put a cross-**X** in column 1,2 and 4 and reconfirm the R1 value in column 3.

The default value is : strong pulldown with R1=85k
Total value of typ. 100kOhm

Option name		strong Pull-Up	weak Pull-Up	R1 value	NO Pull-Up
		1	**2**	**3**	**4**
MPBPU[3]	PB3 input pull-up				
MPBPU[2]	PB2 input pull-up				
MPBPU[1]	PB1 input pull-up				
MPBPU[0]	PB0 input pull-up				

To select an option put a cross-**X** in column 1,2 and 4 and reconfirm the R1 value in column 3.

The default value is : strong pull-up with R1=85k
Total value of typ. 100kOhm

16.2 Digital Watchdog Option

Option name		Default value	user value
		A	**B**
MDigWD	Digital WatchDog	**YES**	

By default the Digital Watchdog is software controlled by the state of the bit **NoLogicWD**. With option MDigWDB the bit **NoLogicWD** has no more effect so the Digital Watchdog is always forced active.

16.3 SWBdataLevel Option

Option name		Default value	user value
		A	**B**
MSWBdataLevel	level of SWB data	**VDD**	

By default the SWB data level (on PPC[1]) will be equal to VDD. With the option **MSWBdataLevelB,** the level of PPC[1] will be selected as VregLogic

16.4 Remaining metal mask options

- For more detail about the mask options refer to the concerned chapters.
- The internal voltage regulators for the logic is adjustable to some extend.
- Different SVLD settings are possible.
- Other changes (functional or parametric) might also be possible using the metal mask.
- If you have a special request please contact EM Microelectronic Marin SA.

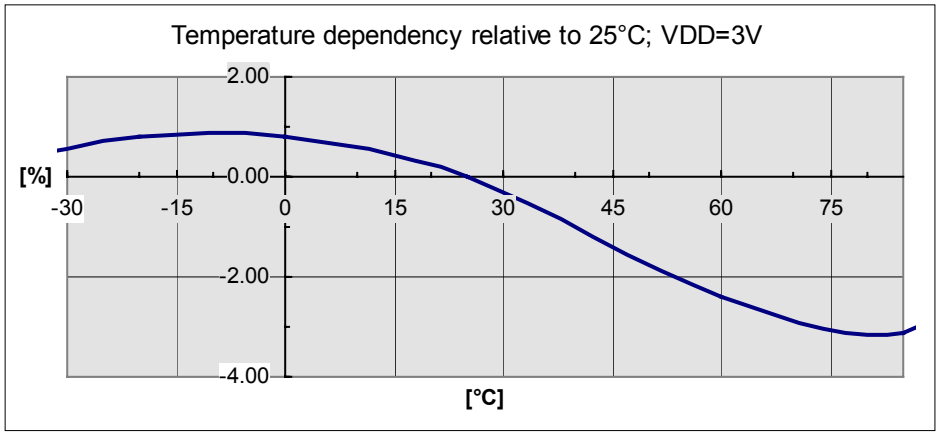
16.5 Metal mask ordering

The customer should specify the required options at the time of ordering.
A copy of the selected option sheet, as well as the « Software ROM characteristic file » generated by the assembler (*.STA) should be attached to the order.

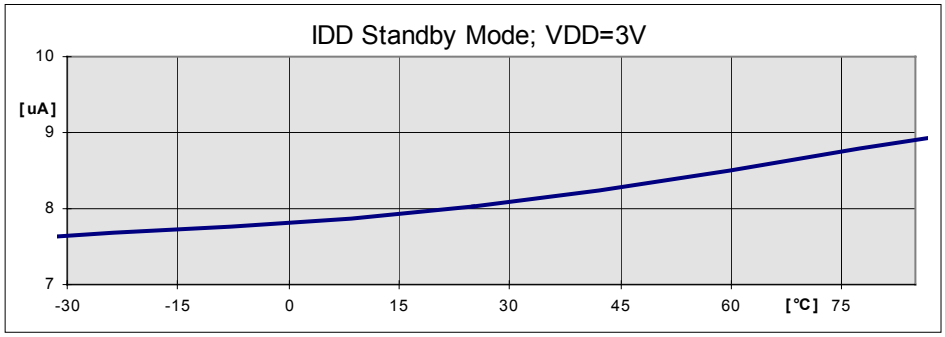
Software name is : _____.bin, dated _____

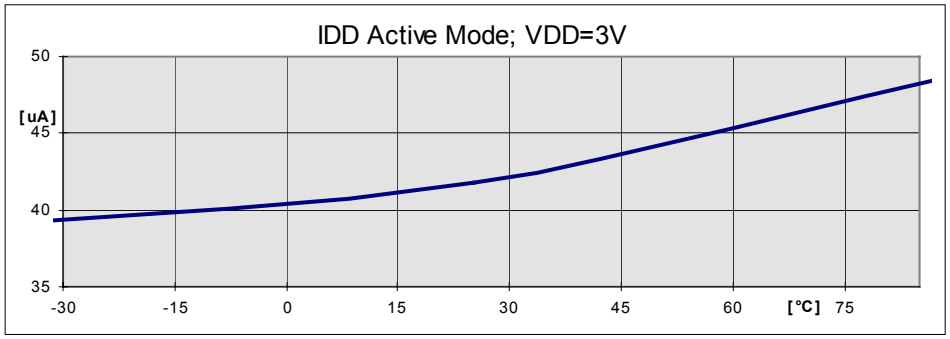
17. Temperature and Voltage Behaviors

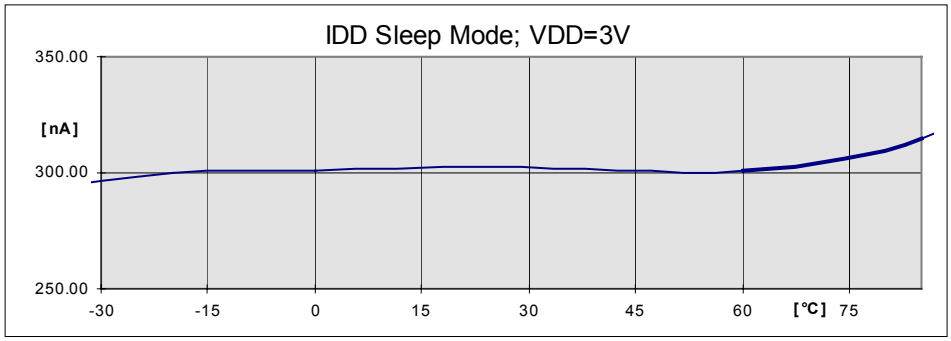
17.1 RC oscillator (typical)



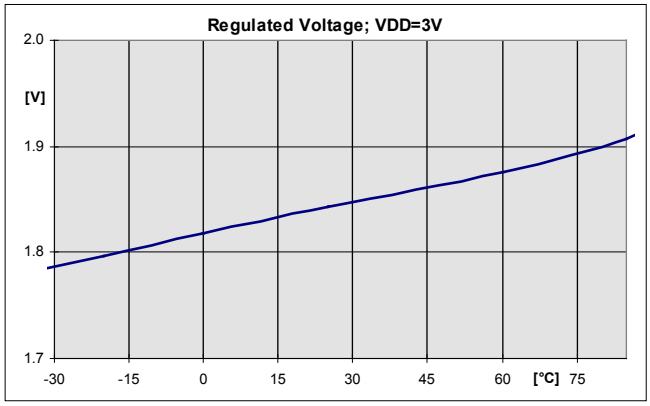
17.2 IDD Current (typical)

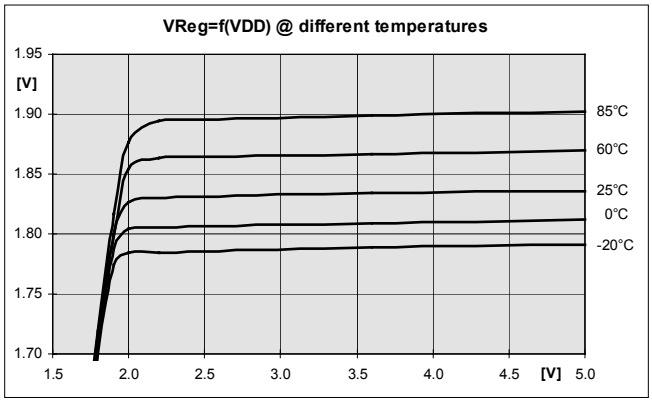




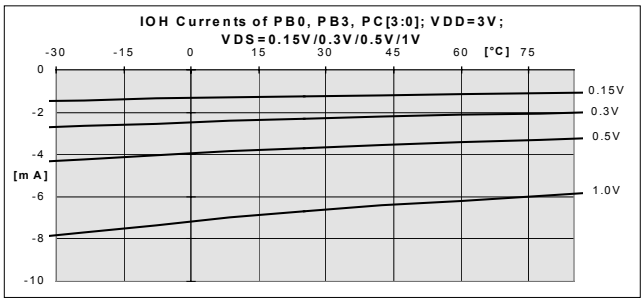


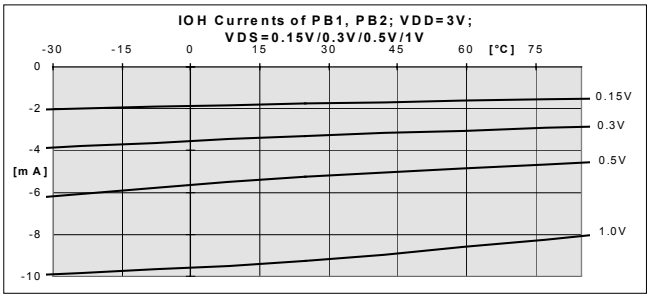
17.3 Regulated Voltage (typical)

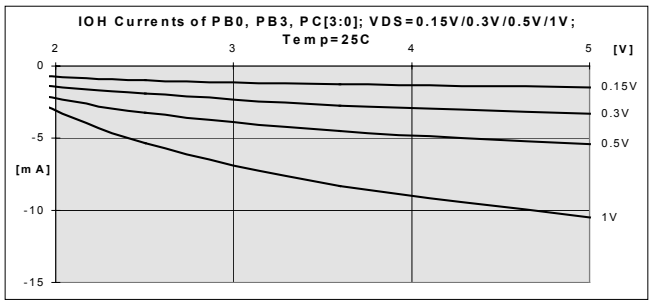


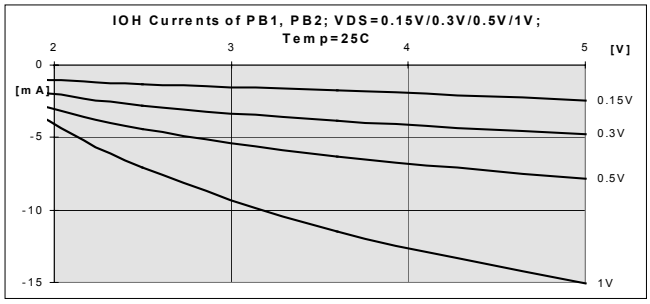


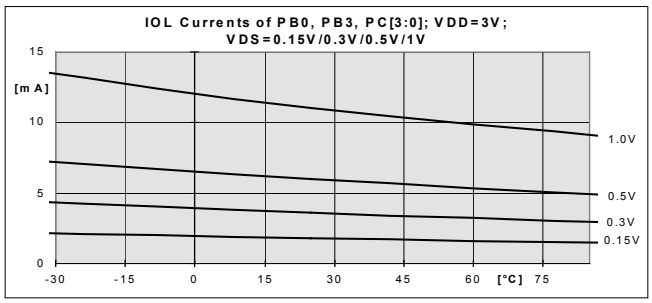
17.4 Output Currents (typical)

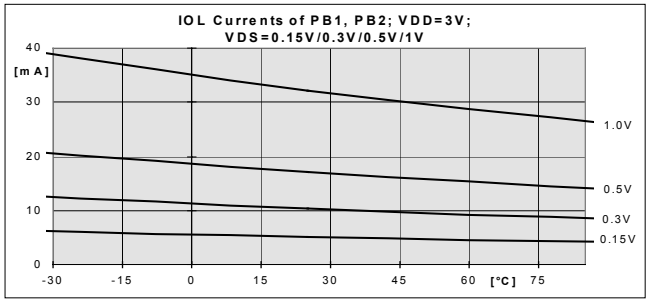


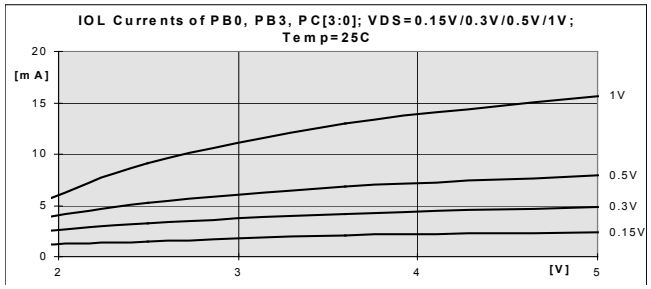


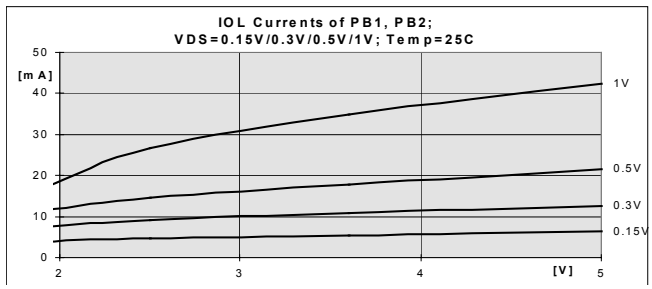




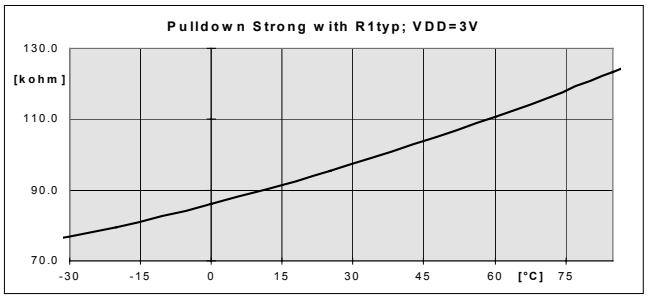


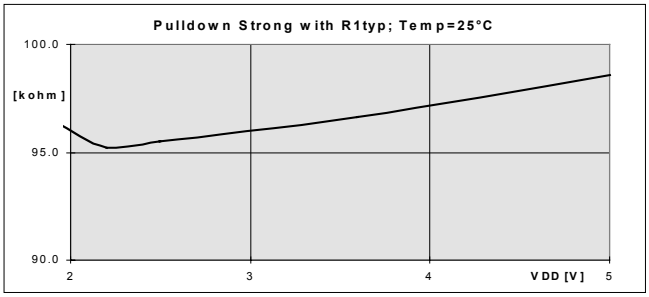


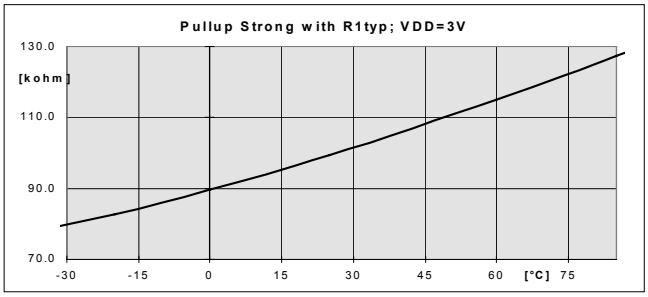


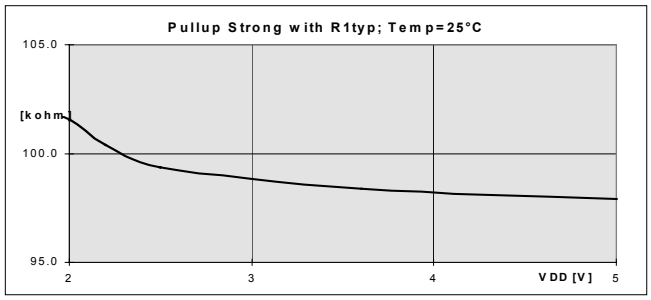


17.5 Pull-up/down (typical)









18. EM6640 Electrical specifications

All electrical specifications written in this section are related to a typical system clock of 600 kHz.

18.1 Absolute maximum ratings

	Min.	Max.	Unit
Power supply VDD-VSS	- 0.2	+ 5.7	V
Input voltage	VSS - 0,2	VDD+0,2	V
Storage temperature	- 40	+ 125	°C
Electrostatic discharge to Mil-Std-883C Method 3015.7 with ref. to VSS	-2000	+2000	V
Maximum soldering conditions		10s x 250°C	

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	-30	25	85	°C	
VDD	1.9	3	5.5	V	
VSS		0		V	Reference terminal
C_{Vreg} (note 1)	0.22	1		µF	regulated voltage capacitor
fq		600		kHz	nominal frequency

Note 1: This capacitor filters switching noise from VDD to keep it away from the internal logic cells. In noisy systems or if using the **MSWBdataLevel** metal option, the capacitor should be chosen bigger than the typical value.

18.3.1 DC characteristics - Power Supply Pins

Conditions: **VDD=3V**, T=25°C, f=600kHz (unless otherwise specified)

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
ACTIVE Supply Current (note 1)		IVDDa		42		µA
	-30 ... 85°C	IVDDa			55	µA
STANDBY Supply Current (in Halt mode)		IVDDh		8		µA
	-30 ... 85°C	IVDDh			12	µA
SLEEP Supply Current (SLEEP = 1)		IVDDs		0.3		µA
	-30 ... 85°C	IVDDs			0.4	µA
POR static level	25°C	VPOR		1.5		V
Regulated voltage	CVreg, no load	Vreg		1.85		V
	-30 ... 85°C,CVreg, no load	Vreg	1.75		2.1	V
RAM data retention		Vrd	1.5			V

Note 1: For test reasons at EM, the user has to provide a test loop with successive writing and reading of two different addresses (5 instructions should be reserved for this measurement).

18.4 Supply Voltage Level Detector

Conditions: Standard operating conditions (unless otherwise specified)

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
SVLD voltage Level	-10 ... 60°C	VSVLD	0.92 VSVLDNom	VSVLDNOM	1.08 VSVLDNom	V
	-30 ... 85°C	VSVLD	0.90 VSVLDNom	VSVLDNOM	1.10 VSVLDNom	V

VSVLDNom = selected SVLD levels by customer (refer to Supply Voltage Level Detector chapter, on page 40)

18.5 Oscillator

Conditions: **Vdd=3V**, T=25°C, f=600kHz (unless otherwise specified), f_O = f at 25°C

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
Temperature stability	-30°C ... +40°C	df/f_O		±1		%
Temperature stability	-30°C	df/f_O			+3	%
Temperature stability	85°C	df/f_O	-6			%
Adjustable frequency range permitted		freq	510	600	690	kHz
Delivery state			590	600	610	kHz
Oscillator start time	VDD > VDDmin	tdosc		50		µs
System start time (oscillator + cold start + reset)		tdsys		1	5	ms
Oscillation detector frequency	VDD > VDDmin	f_{OD}		100		kHz

18.6 Analogue filter on PortA

Conditions: Standard operating conditions (unless otherwise specified)

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
time Constant		anafltr	5	9	15	µs

18.7 Sleep counter reset (SCR)

Conditions: Standard operating conditions (unless otherwise specified)

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
time Constant		ctscr	7	10	13	ms

18.8 EEPROM

Conditions: Standard operating conditions (unless otherwise specified)

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
Read time (note 1)	-30 ... 85°C	EEPrd		13		µs
Write time (note 1)	-30 ... 85°C	EEPwr		20.2		ms
VDD during write and read operation	-30 ... 85°C	VEEP	2.0		5.5	V

Note 1 : This values are guaranteed by design when using 600kHz.

18.9 DC characteristics - input / output Pins

Conditions: Standard operating conditions (unless otherwise specified)

Parameter	Conditions	Symb	Min.	Typ.	Max.	Unit
Input Low voltage						
Ports A,B,C	Input Mode	VIL	Vss		0.3VDD	V
TEST			Vss		0.3VDD	V
Input High voltage						
Ports A,B,C	Input Mode	VIH	0.7VDD		VDD	V
TEST			0.7VDD		VDD	V
Output Low current						
Port B[3], Port B[0]	VDD=1.9V , VOL=0.30V	IOL		2.5		mA
	VDD=3.0V , VOL=0.30V	IOL	2.6	3.6		mA
	VDD=5.0V , VOL=0.30V	IOL		5		mA
Port B[2], Port B[1]	VDD=1.9V , VOL=0.30V	IOL		7.5		mA
	VDD=3.0V , VOL=0.30V	IOL	6.4	10.2		mA
	VDD=5.0V , VOL=0.30V	IOL		13		mA
Port C[3:0]	VDD=1.9V , VOL=0.30V	IOL		2.5		mA
	VDD=3.0V , VOL=0.30V	IOL	2.6	3.6		mA
	VDD=5.0V , VOL=0.30V	IOL		5		mA
Port C[1] (note 1)	VDD≥2.2V , VOL=0.10V	IOL	2.8	3.7		mA
Output High current						
Port B[3], Port B[0]	VDD=1.9V, VOH= VDD-0.30V	IOH		-1.4		mA
	VDD=3.0V, VOH= VDD-0.30V	IOH		-2.5	-1.9	mA
	VDD=5.0V, VOH= VDD-0.30V	IOH		-3.3		mA
Port B[2], Port B[1]	VDD=1.9V, VOH= VDD-0.30V	IOH		-1.8		mA
	VDD=3.0V , VOH= VDD-0.30V	IOH		-3.3	-2.6	mA
	VDD=5.0V , VOH= VDD-0.30V	IOH		-4.8		mA
Port C[3:0]	VDD=1.9V , VOH= VDD-0.30V	IOH		-1.4		mA
	VDD=3.0V , VOH= VDD-0.30V	IOH		-2.5	-1.9	mA
	VDD=5.0V , VOH= VDD-0.30V	IOH		-3.3		mA
Port C[1] (note 1)	VDD≥2.2V, VOH=VregLog-0.1v	IOH		-2	-1.25	mA

Note 1 : Depending on the Mask Option sheet. By default the SWB data level (on PPC[1]) will be equal to VDD. With option **MSWBdataLevelB,** the level of PPC[1] will be selected as VregLogic

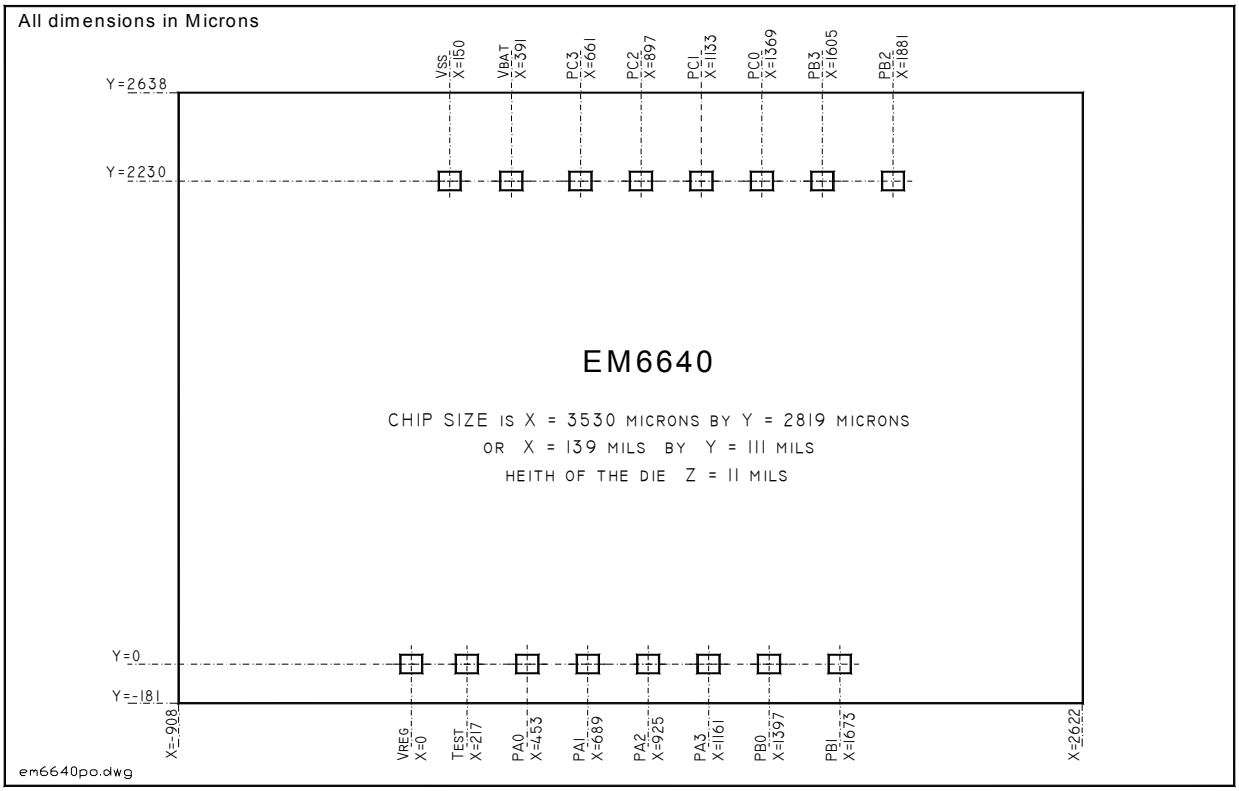
18.10 DC characteristics - pull up/down

Conditions: T=25°C (unless otherwise specified)

Parameter	Conditions	Symb	Min.	Typ.	Max.	Unit
Input Pull-down	VDD=1.9V, Pin at 1.9V	R_{PD}		15k		Ohm
Test	VDD=3.0V, Pin at 3.0V	R_{PD}	9K	15k	22K	Ohm
	VDD=5.0V, Pin at 5.0V	R_{PD}		15k		Ohm
Input Pull-down	VDD=1.9V, Pin at 1.9V, weak	R_{PD}		200k		Ohm
Port A,B,C (note 1)	VDD=3.0V, Pin at 3.0V, weak	R_{PD}		300k		Ohm
	VDD=5.0V, Pin at 5.0V, weak	R_{PD}		450k		Ohm
Input Pull-up	VDD=1.9V, Pin at 0.0V, weak	R_{PU}		350k		Ohm
Port A,B,C (note 1)	VDD=3.0V, Pin at 0.0V, weak	R_{PU}		200k		Ohm
	VDD=5.0V, Pin at 0.0V, weak	R_{PU}		150k		Ohm
Input Pull-down	VDD=1.9V, Pin at 1.9V, strong	R_{PD}		97k		Ohm
Port A,B,C (note 1)	VDD=3.0V, Pin at 3.0V, strong	R_{PD}	70k	96k	130k	Ohm
	VDD=5.0V, Pin at 5.0V, strong	R_{PD}		98k		Ohm
Input Pull-up	VDD=1.9V, Pin at 0.0V, strong	R_{PU}		102		Ohm
Port A,B,C (note 1)	VDD=3.0V, Pin at 0.0V, strong	R_{PU}	70k	97k	130k	Ohm
	VDD=5.0V, Pin at 0.0V, strong	R_{PU}		96k		Ohm

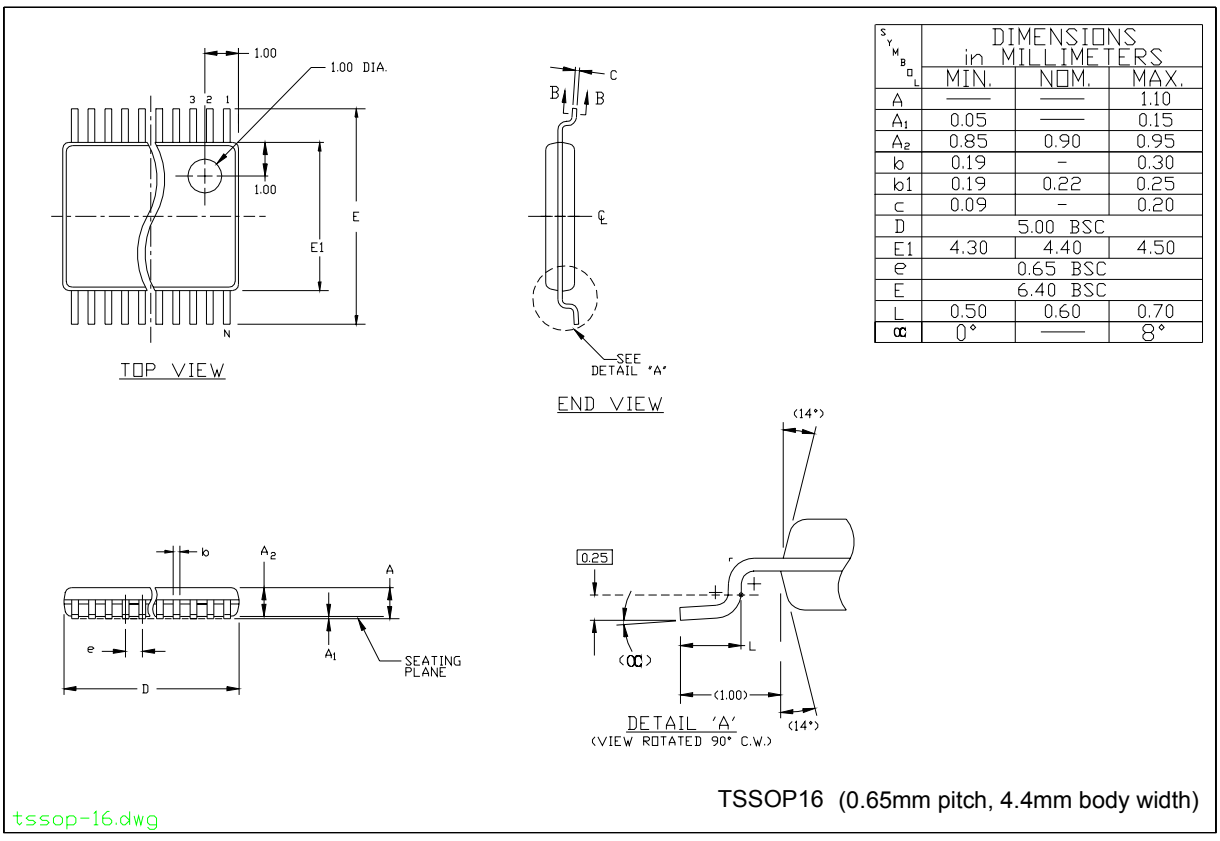
Note 1 : Weak or strong are standing for weak or strong pull transistor. Values are for R1typ=85kOhm.

19. Pad Location Diagram



20. PACKAGE & Ordering Information

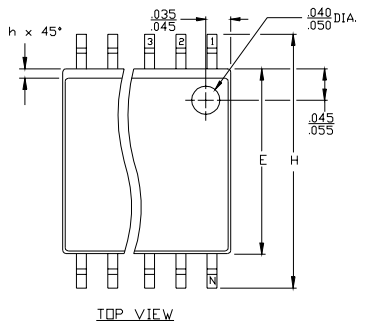
TSSOP 16



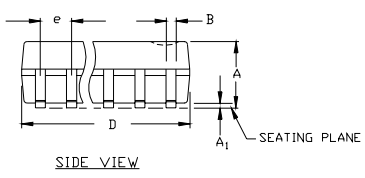
SYMBOL	DIMENSIONS in MILLIMETERS		
	MIN.	NOM.	MAX.
A	——	——	1.10
A_1	0.05	——	0.15
A_2	0.85	0.90	0.95
b	0.19	–	0.30
b1	0.19	0.22	0.25
c	0.09	–	0.20
D	5.00 BSC		
E1	4.30	4.40	4.50
e	0.65 BSC		
E	6.40 BSC		
L	0.50	0.60	0.70
α	0°	——	8°

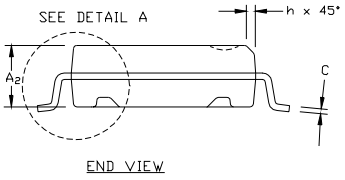
TSSOP16 (0.65mm pitch, 4.4mm body width)

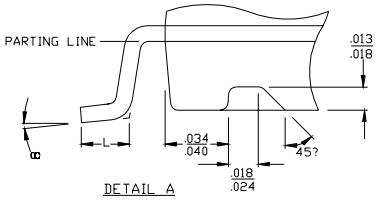
SOP 16



SYMBOL	DIMENSIONS in INCHES			DIMENSIONS in MILLIMETERS		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	.097	.101	.104	2.46	2.56	2.64
A_1	.0050	.009	.0115	0.127	0.22	0.29
A_2	.090	.092	.094	2.29	2.34	2.39
B	.014	.016	.019	0.35	0.41	0.48
C	.0091	.010	.0125	0.23	0.25	0.32
D	.402	.407	.412	10.21	10.34	10.46
E	.292	.296	.299	7.42	7.52	7.59
e		.050 BSC			1.27 BSC	
H	.400	.406	.410	10.16	10.31	10.41
h	.010	.013	.016	0.25	0.33	0.41
L	.024	.032	.040	0.61	0.81	1.02
N		16			16	
α	0°	5°	8°	0°	5°	8°



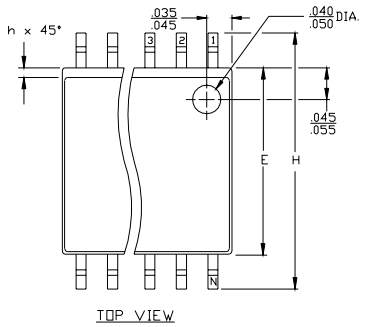




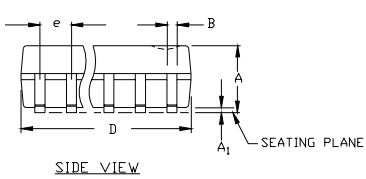
Sop-16.dwg

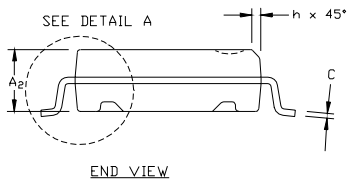
SOP-16(1.27mm pitch, 300mils body width)

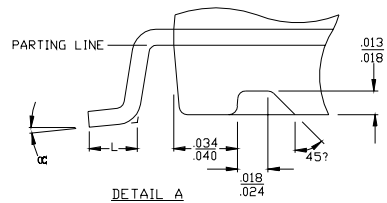
SOP 18



SYMBOL	DIMENSIONS in INCHES			DIMENSIONS in MILLIMETERS		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	.097	.101	.104	2.46	2.56	2.64
A_1	.0050	.009	.0115	0.127	0.22	0.29
A_2	.090	.092	.094	2.29	2.34	2.39
B	.014	.016	.019	0.35	0.41	0.48
C	.0091	.010	.0125	0.23	0.25	0.32
D	.451	.456	.461	11.46	11.58	11.71
E	.292	.296	.299	7.42	7.52	7.59
e		.050 BSC			1.27 BSC	
H	.400	.406	.410	10.16	10.31	10.41
h	.010	.013	.016	0.25	0.33	0.41
L	.024	.032	.040	0.61	0.81	1.02
N		18			18	
α	0°	5°	8°	0°	5°	8°







Sop-18.dwg

SOP-18 (1.27mm pitch, 300mils body width)

NB : Pin N°17 & N°18 are not connected.
This package is fully compatible with the MFP version (EM6540)

20.1 Ordering Information

Packaged Device:

EM6640 %%% TP16 B

Customer Version:
customer-specific number
given by EM Microelectronic

Package:
SO16, SO18 = 16/18 pin SOIC
TP16 = 16 pin TSSOP

Delivery Form:
A = Stick
B = Tape&Reel

Device in DIE Form:

EM6640 %%% WS 11

Customer Version:
customer-specific number
given by EM Microelectronic

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)

Ordering Part Number (selected examples)

Part Number	Package/Die Form	Delivery Form/ Thickness
EM6640%%%TP16A	16 pin TSSOP	Stick
EM6640%%%TP16B	16 pin TSSOP	Tape&Reel
EM6640%%%SO18B	18 pin SOIC	Tape&Reel
EM6640%%%WS11	Sawn wafer	11 mils
EM6640%%%WP11	Die in waffle pack	11 mils

Please make sure to give the complete Part Number when ordering, including the 3-digit version. The version is made of 3 digits %%%: the first one is a letter and the last two are numbers, e.g. P04 , P12, etc.

20.2 Package Marking

SOIC marking:

First line:	E	M	6	6	4	0		0	%	%	Y
Second line:	P	P	P	P	P	P	P	P	P	P	P
Third line:	C	C	C	C	C	C	C	C	C	C	C

TSSOP marking:

6	6	4	0	%	%
P	P	P	P	P	P
		P	P	P	Y

Where: %% = last two-digits of the customer-specific number given by EM (e.g. 04, 12, etc.)
Y = Year of assembly
PP…P = Production identification (date & lot number) of EM Microelectronic
CC…C = Customer specific package marking on third line, selected by customer

20.3 Customer Marking

There are **11** digits available for customer marking on **SO16/18**.
There are **no** digits available for customer marking on **TSSOP16**.

Please specify below the desired customer marking.

21. Updates of specifications

Revision	Date of Update Name	Chapter concerned	Old Version (Text, Figure, etc.)	New Version (Text, Figure, etc.)
A/220	16.6.98	ALL		Spelling Corrections, Figure updates, added Die size and Package drawings, added graphs
B/391	01/11/01 PERT	All	-	Change heater & footer Add URL.
C/446	24/03/02 PERT	62/64	-	Change Pad Loc. Diagram & ordering information