Ultra Low Power Microcontroller with ADC AND EEPROM

Features

- Low Power - 3.2 µA active mode, ADC off
  - 9.0 µA active mode, ADC on
  - 0.6 µA standby mode
  - 0.1µA sleep mode
    @ 3.0V, 32kHz, 25°C
- Voltage range logic incl. EEPROM 2.0 to 5.5 V
- System operating clock : 32 or 128KHz (metal option)
- Voltage range for the ADC is 2.6 to 5.5 V
- 2 clocks per instruction cycle
- 72 basic instructions
- ROM 3k × 16 bit
- RAM 128 × 4 bit
- E^2PROM 64 × 8 bit
- Voltage Level Detector, 3 levels software selectable :2.2, 2.5, 3.0 V
- 2 channel ADC, successive approximation method; conversion time at 32 kHz : 305µs
- Max. 12 inputs (3 ports); port A, port B, port C
- Max. 8 outputs (2 ports); port B, port C
- Serial Write Buffer, 256 bit wide , 4 bit rates
- Oscillation supervisor and timer watchdog
- Universal 10-bit counter, PWM, event counter
- 8 internal interrupt sources (2 × timer , 2 × prescaler, ADC, VLD, FIFO, EEPROM)
- 4 external interrupt sources (input port A)
- Frequency output; 32kHz, 2kHz, 1kHz, PWM

Description

The EM6617 is an advanced single chip CMOS 4-bit microcontroller. It contains ROM, RAM, power on reset, watchdog timer, oscillation detection circuit, combined timer , event counter, prescaler, E^2PROM, 2 channel ADC, serial write buffer, voltage level detector and several clock functions. The low voltage feature and low power consumption make it the most suitable controller for battery, stand alone and mobile equipment. The EM6617 is manufactured using EM Microelectronic's advanced low power (ALP) CMOS Process.

Typical Applications

- Sensor & detector interface
- Heat meter interface
- Security systems
- Household equipment controls
- Automotive controls
- Measurement equipment
- R/F and IR. control
- Voltage control

Figure 1. Architecture

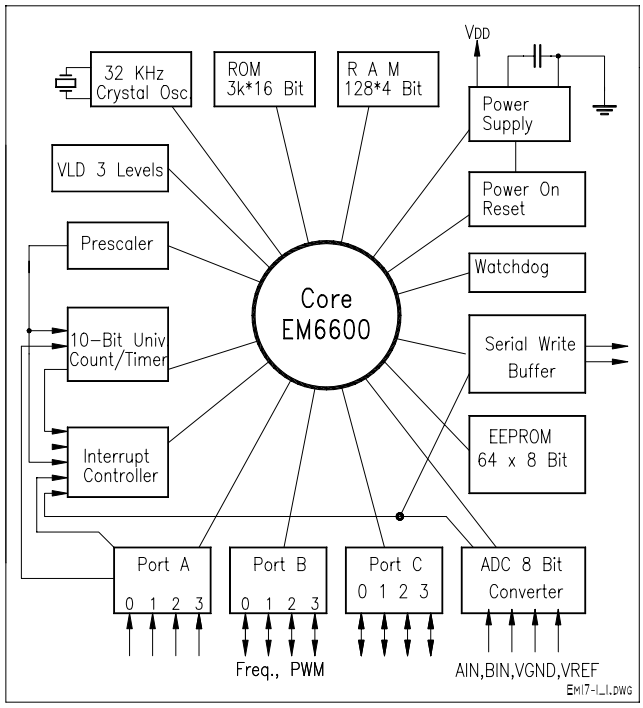
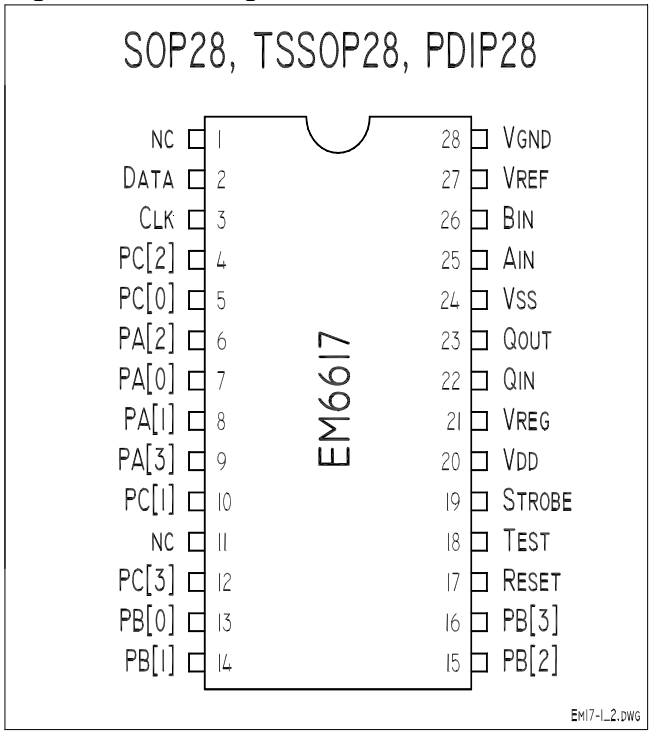


Figure 2. Pin Configuration



EM6617 at a glance

• Power Supply
- Low voltage low power architecture including internal voltage regulator
- 2.0 ... 5.5 V battery voltage for all logic functions
- 2.6 ... 5.5 V battery voltage for the ADC (0.2LSB)
- 3.2 µA in active mode, ADC off
- 9.0 µA active mode, ADC on
- 0.6µA in standby mode
- 0.1µA in sleep mode
- 32 KHz crystal oscillator

• RAM
- 64 x 4 bit, direct addressable
- 64 x 4 bit, indirect addressable

• ROM
- 3072 x 16 bit metal mask programmable

• E^2PROM
- 64 x 8 bit, indirect addressable
- Interrupt request at the end of a write operation

• CPU
- 4 bit RISC architecture
- 2 clock cycles per instruction
- 72 basic 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 resets (logic and oscillation watchdogs)
- Reset terminal
- Reset with input combination on port A register selectable, ¨AND¨ or ¨OR¨ type by metal mask

• 4-Bit Input Port A
- Direct input read on the port terminals
- Debouncer function available on all inputs
- Interrupt request on positive or negative edge
- Pull-up or pull-down or none selectable by register
- Test variables (software) for conditional jumps
- PA[0] and PA[3] are inputs for the event counter
- Reset with input combination (register selectable)

• Serial Write Buffer (output)
- Max 256 bits long bit rates of 16kHz,8kHz,2kHz,1kHz
- Automatic or interactive send mode
- Interrupt request when buffer is empty

• 2 Channel 8-bit ADC
- Conversion time is 305µs @32kHz
- 2 operating modes (continuous, single)
- Interrupt request at the end of conversion

• Prescaler
- 15 stage system clock divider down to 1 Hz
- 2 Interrupt requests; 1 Hz, 32 Hz or 8 Hz
- Prescaler reset (4 KHz to 1Hz)

• 4-Bit Bi-directional Port B
- All different functions bit-wise selectable
- Direct input read on the port terminals
- Data output latches
- CMOS or Nch. open drain outputs
- Pull-down or pull-up selectable
- Weak pull-up in Nch. open drain mode
- Selectable PWM, 1kHz, 32kHz and 2kHz output

• 4Bit Bi-directional Port C
- Input or output mode as whole port
- Direct input read on port terminal
- Data output latches
- CMOS or Nch. open drain outputs
- Pull-down or pull-up selectable
- Weak pull-up in Nch. open drain mode

• Voltage Level Detector
- 3 levels software selectable (2.0, 2.5, 3.0 V)
- Busy flag during measure
- Interrupt request at end of measure

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

• Interrupt Controller
- 4 external and 8 internal interrupt request sources
- Each interrupt can individually be maskable
- Each interrupt can individually be reset
- Automatic reset of each interrupt request after read
- General interrupt request to CPU can be disabled
- Automatic enabling of general interrupt request flag when going into HALT mode

Table of Contents

1. Pin Description for EM6617

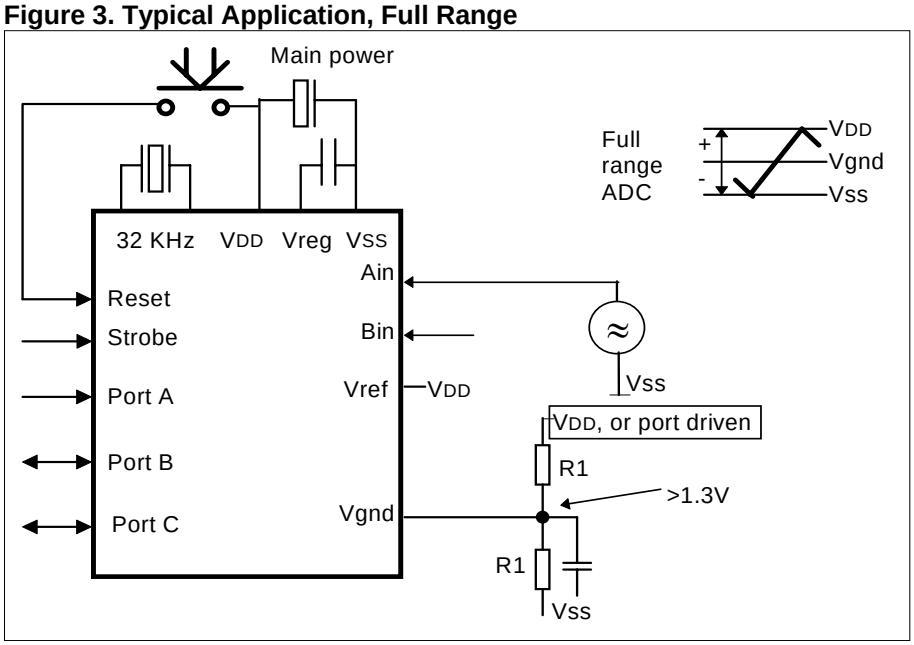
PDIP24 SO24	PDIP28 SO28 TSSOP28	Signal Name	Function	Remarks
17	20	VBAT=VDD	Positive power supply	Main power pin MFP programming connection
21	24	Vss	Negative power supply	Reference terminal, substrate MFP programming connection
18	21	Vreg	Internal voltage regulator	connect to minimum 100nF MFP programming connection
15	18	Test	Input test terminal, internal pull-down 15k	for EM tests only, ground 0 ! Except for MFP programming
14	17	Reset	Reset terminal internal pull-down 15k	
16	19	Strobe	Strobe / reset status	µC reset state + port B write
19	22	Qin	Crystal terminal 1	32kHz crystal MFP programming connection
20	23	Qout	Crystal terminal 2	32kHz crystal MFP programming connection
10	13	PB[0]	Input or output, CMOS or Nch. open drain; port B terminal 0	Ck[12] output (2 KHz)
11	14	PB[1]	Input or output, CMOS or Nch. open drain; port B terminal 1	Ck[16] output (32 KHz)
12	15	PB[2]	Input or output, CMOS or Nch. open drain; port B terminal 2	Ck[11] output (1 KHz)
13	16	PB[3]	Input or output, CMOS or Nch. open drain; port B terminal 3	PWM output
6	7	PA[0]	Input port A terminal 0	TestVar 1, event counter
7	8	PA[1]	Input port A terminal 1	TestVar 2
5	6	PA[2]	Input port A terminal 2	
8	9	PA[3]	Input port A terminal 3	Event counter
4	5	PC[0]	Input or output, CMOS or Nch. open drain; port C terminal 0	
9	10	PC[1]	Input or output, CMOS or Nch. open drain; port C terminal 1	
-	4	PC[2]	Input or output, CMOS or Nch. open drain; port C terminal 2	Bonded only in 28 pin package
-	12	PC[3]	Input or output, CMOS or Nch. open drain; port C terminal 3	Bonded only in 28 pin package
22	25	Ain	channel A for A/D converter	
23	26	Bin	channel B for A/D converter	
24	27	Vref	external voltage reference input FOR the A/D converter	Only used for external Vref i.e. Vref not equal to VDD
1	28	Vgnd	Virtual analogue ground for A/D converter	Virtual Ground, usually VDD/2
2	2	Data	Serial write buffer data out	
3	3	Clk	Serial write buffer clock out	

Gray shaded area : MFP programming connections (VDD, Vreg, Qin , Qout, Test, Vss).

2. Typical configurations

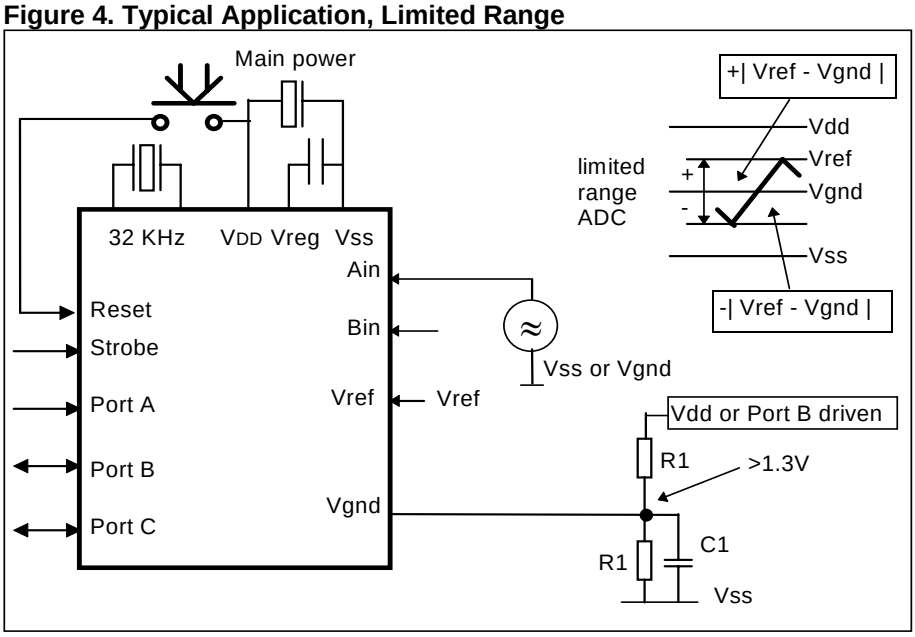
Full range ADC : Vref = VDD, Vgnd = VDD/2.
For power saving one might connect the Vgnd resistor divider chain onto a port B output. This output should be driving VDD during the conversion and driving VSS or high impedance in the ADC off state.

Figure 3. Typical Application, Full Range



Limited range ADC : VDD > Vref > Vgnd, Vgnd=VDD/2.
For power saving one might connect the Vgnd and the Vref resistor divider chain onto a port B output to VSS. This output should be driving VDD during the conversion and driving Vss or high impedance in the ADC off state.

Figure 4. Typical Application, Limited Range



other possibility: VREF = VregLogic, VGND = VregLogic/2
For power saving one might connect the Vgnd resistor divider chain from VregLogic onto a port B output. This output should be driving VSS during the conversion and driving 'high impedance' in the ADC off state.

3. Operating Modes

The EM6617 has two low power dissipation modes, standby and sleep. Figure 5 is a transition diagram for these modes.

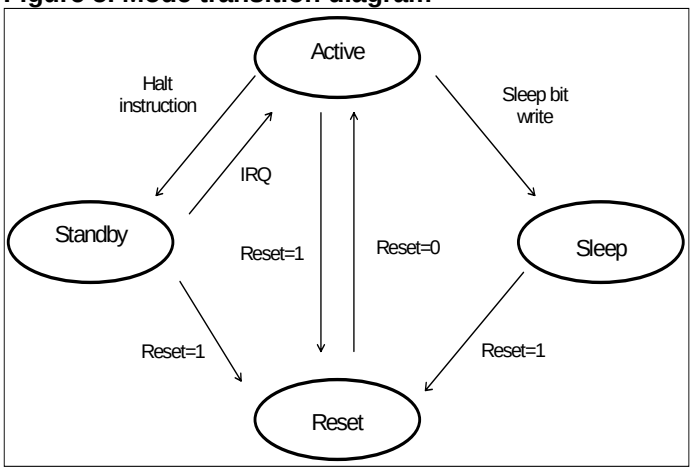
3.1 Active Mode

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

3.2 Standby Mode

Executing a halt instruction puts the EM6617 into standby mode. The voltage regulator, oscillator, watchdog timer, ADC, interrupts, SWB, timers and counters are operating. However, the CPU stops since the clock related to instruction execution stops. Registers, RAM and I/O pins retain their states prior to standby mode. A reset or an interrupt request if enabled cancels standby.

Figure 5. Mode transition diagram



3.3 Sleep Mode

Writing to the **Sleep** bit in the **RegSysCntl1** register puts the EM6617 in sleep mode. The oscillator stops and most functions of the EM6617 are inactive. To be able to write to the **Sleep** bit, the **SleepEn** bit in **RegSysCntl2** must first be set to "1". In sleep mode only the voltage regulator and the reset input are active. The RAM data integrity is maintained. Sleep mode may be canceled only by a high level of min 10µs at the Reset terminal or by the selected port A input reset combination, if option **InpResSleep** in register **OPTFSelPB** is turned on.

Due to the cold-start characteristics of the oscillator, waking up from sleep mode may take some time to guarantee stable oscillation. During sleep mode and the following start up the EM6617 is in reset state. Waking up from sleep clears the **Sleep** flag but not the **SleepEn** bit. Inspecting the **SleepEn** allows to determine if the EM6617 was powered up (**SleepEn** = "0") or woken up from sleep (**SleepEn** = "1").

Table 3.3.1. Internal State in Standby and Sleep Mode

Function	Standby	Sleep
Oscillator	Active	Stopped
Oscillator Watchdog	Active	Stopped
Instruction Execution	Stopped	Stopped
Interrupt Functions	Active	Stopped
Registers and Flags	Retained	Reset
RAM Data	Retained	Retained
Option Registers	Retained	Retained
Timer & Counter	Active	Reset
Logic Watchdog	Active	Reset
I/O Port B and Serial Port	Active	High Impedance, Pull's as defined in option register
Input Port A	Active	No pull-downs and inputs deactivated except if **InpResSleep** = "1"
LCD	Active	Stopped (display off)
Strobe Output	Active	Active
Buzzer Output	Active	High Impedance
Voltage Level Detector	Finishes ongoing measure, then stop	Stopped
Reset Pin	Active	Active

4. Power Supply

The EM6617 is supplied by a single external power supply between VDD (Vbat) and VSS (ground). A built-in voltage regulator generates Vreg providing regulated voltage for the oscillator and the internal logic. The output drivers and the ADC are supplied directly from the external supply VDD. A typical power connection configuration and the internal power connection is shown below.

Figure 6. Typical Power Connection

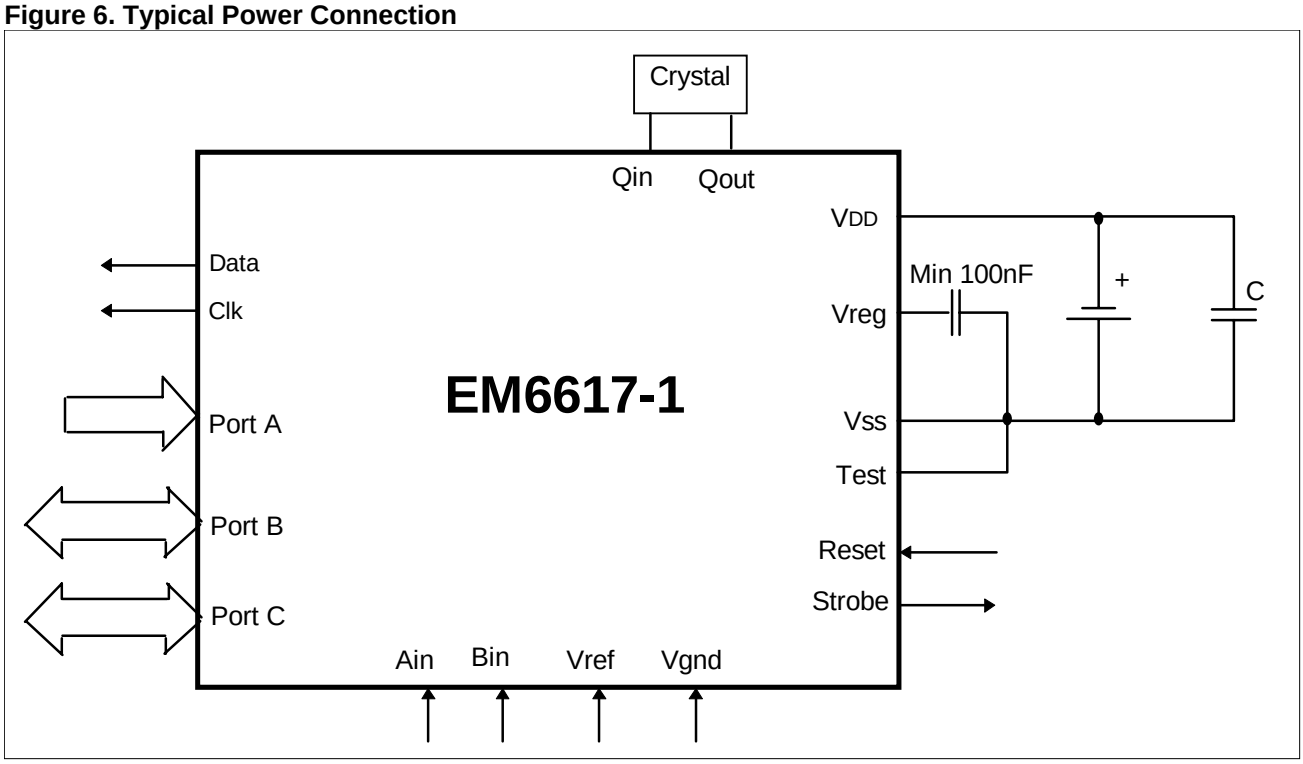
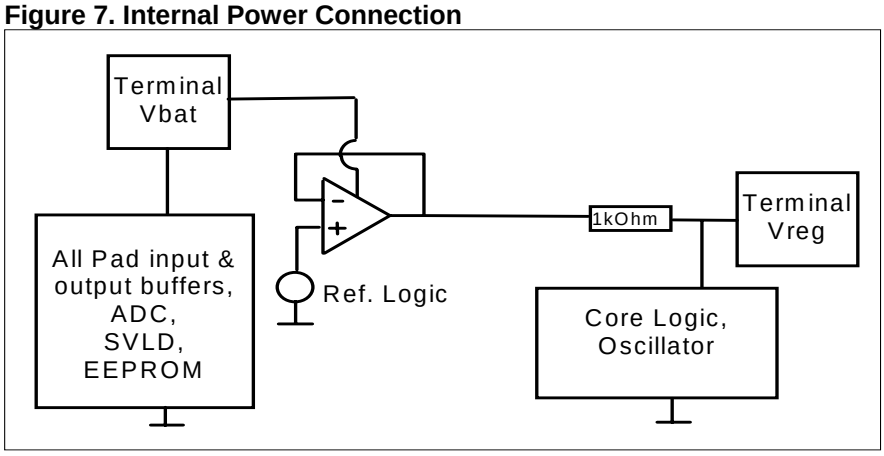


Figure 7. Internal Power Connection



5. Reset

Figure 8. illustrates the reset structure of the EM6617-1. There are six possible reset sources :

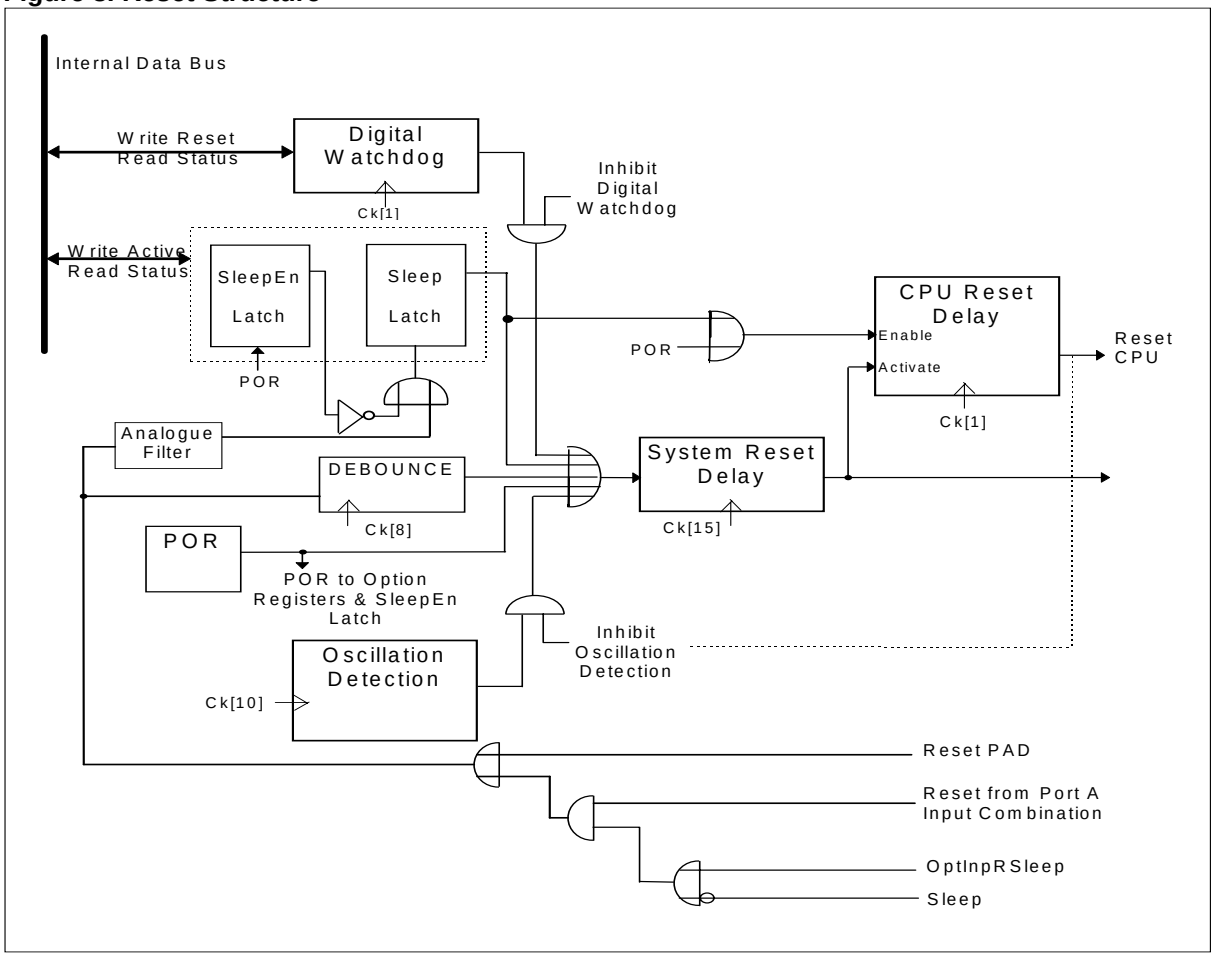
(1) Internal initial reset from the Power On Reset (POR) circuitry. --> POR
(2) External reset from the Reset terminal. --> System Reset, Reset CPU
(3) External reset by simultaneous high/low inputs to port A. --> System Reset, Reset CPU
(Combinations are defined in the registers **OptInpRSel1** and **OptInpRSel2)**
(4) Internal reset from the Digital Watchdog. --> System Reset, Reset CPU
(5) Internal reset from the Oscillation Detection Circuit. --> System Reset, Reset CPU
(6) Internal reset when sleep mode is activated. --> System Reset, Reset CPU

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 system clock 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 (**SleepEn)** latch. System reset and reset CPU do not reset the option registers nor the **SleepEn** latch.

Reset state can be shown on Strobe terminal by selecting **StrobeOutSel1,0 = 0** in **OPTCandStr** register.

Figure 8. Reset Structure



5.1 Oscillation Detection Circuit

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 EM6617 remains in the reset state for the 'CPU Reset Delay', to allow the oscillator to stabilize after power up.
The oscillator is disabled during sleep mode. So when waking up from sleep mode, the CPU of the EM6617 remains in the reset state for the CPU Reset Delay, to allow the oscillator to stabilize. During this time, the Oscillation Detection Circuit is inhibited.

In active or standby modes, the oscillator detection circuit monitors the oscillator. If it stops for any reason, a system reset is generated. After clock restart the CPU waits for the CPU Reset Delay before executing the first instructions.
The oscillation detection circuitry can be inhibited with bit **NoOscWD** = 1 in register **RegSysCntl3.** At power up, and after any system reset, the function is activated.

The 'CPU Reset Delay' is 32768 system clocks (Ck[16]) long.

5.2 Reset Terminal

During active or standby modes the Reset terminal has a debouncer to reject noise. Reset must therefore be active for at least 16 ms (system clock = 32 KHz).
When canceling sleep mode, the debouncer is not active (no clock), however, reset passes through an analogue filter with a time constant of typical. 5µs. In this case Reset pin must be high for at least 10 µs to generate a system reset.

5.3 Input Port A Reset Function

By writing the **OptInpRSel1** and **OptInpRSel2** registers it is possible to choose any combination of port A input values to execute a system reset. The reset condition must be valid for at least 16ms (system clock = 32kHz) in active and standby mode.
OPTInpRSleep selects the input port A reset function in sleep mode. If set to "1" the occurrence of the selected combination for input port A reset will immediately trigger a system reset (no debouncer) .
Reset combination selection (*InpReset)* is done with registers **OptInpRSel1** and **OptInpRSel2.**
Either an 'AND' or an 'OR' type port A combination can be chosen to generate the reset.

5.3.1 AND-Type Reset function

Default setting(metal option). One or a combination of port A inputs will trigger a reset. Following formula is applicable :

InpResPA = **InpResPA[0] • InpResPA[1] • InpResPA[2] • InpResPA[3]**

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

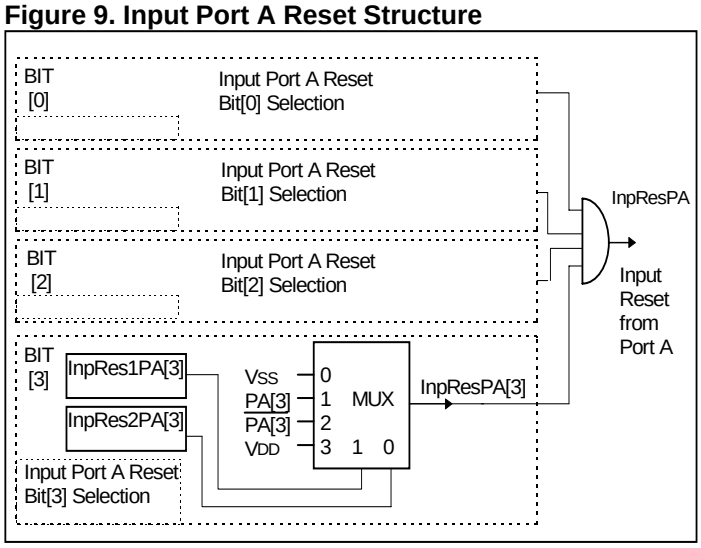
n = 0 to 3

i.e. ; - no reset if InpResPA[n] = VSS.

- Don't care function on a single bit with its InpResPA[n] = VDD.

- Always Reset if InpResPA[3:0] = 'b1111

Figure 9. Input Port A Reset Structure



5.3.2 OR -Type Reset function

If wanted, needs to be chosen with the metal 1 option settings. Any one of the port A inputs can trigger a reset. Following formula is applicable :

InpResPA = **InpResPA[0]** + **InpResPA[1]** + **InpResPA[2]** + **InpResPA[3]**

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

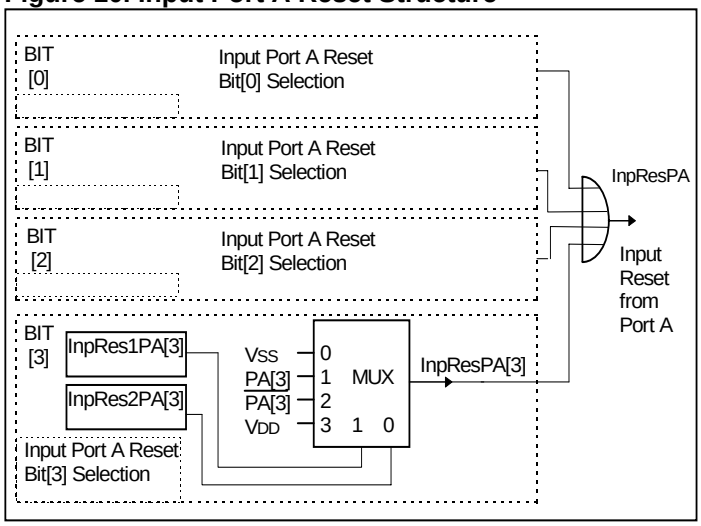
n = 0 to 3

i.e. ; - no reset if all InpResPA[n] = VSS.

- Don't care function on a single bit with its InpResPA[n] = Vss.

- Always Reset if any InpResPA[3:0] = VDD

Figure 10. Input Port A Reset Structure



5.4 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 **RegSysCntl3**. 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 = 32 KHz), or a system reset signal is generated.

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

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

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

Table 5.4.1 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 Ck[1] divided by 4
0	WDVal0	0	R	Watchdog timer data Ck[1] divided by 2

Table 5.4.2 Watchdog Control Register RegSysCntl3

Bit	Name	Reset	R/W	Description
3	*Vref1/2Sel*	*0*	*R/W*	*Reference selection for the ADC*
2	--	*0*	*R/W*	*always reads 0*
1	NoOscWD	0	R/W	No oscillation supervisor
0	NoLogicWD	0	R/W	No logic watchdog

5.5 CPU State after Reset

Reset initializes the CPU as shown in Table 5.5.1 below.

Table 5.5.1 Initial CPU Value after Reset.

Name	Bits	Symbol	Initial Value
Program counter 0	12	PC0	hex 000 (as a result of Jump 0)
Program counter 1	12	PC1	Undefined
Program counter 2	12	PC2	Undefined
Stack pointer	2	SP	PSP[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

6. Oscillator and Prescaler

6.1 Oscillator

A built-in crystal oscillator generates the system operating clock for the CPU and peripheral blocks, from an externally connected crystal (typically 32.768kHz or 128KHz depending of the metal opt. on table 19.1.6). The oscillator circuit is supplied by the regulated voltage, Vreg. In sleep mode the oscillator is stopped.
EM's special design techniques guarantee the low current consumption of this oscillator. The external impedance between the oscillator pads must be greater than 10MOhm. Connection of any other components to the two oscillator pads must be confirmed by EM Microelectronic-Marin SA.

6.2 Prescaler

The prescaler consists of fifteen elements divider chain which delivers clock signals for the peripheral circuits such as timer/counter, buzzer, LCD voltage multiplier, debouncer and edge detectors, as well as generating prescaler interrupts. The input to the prescaler is the system clock signal. Power on initializes to Hex(0001).

Table 6.2.1 Prescaler Clock Name Definition

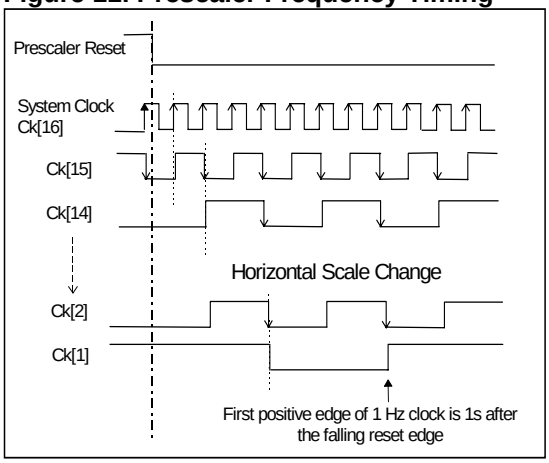
Function	Name	32 KHz Xtal
System clock	Ck[16]	32768 Hz
System clock / 2	Ck[15]	16384 Hz
System clock / 4	Ck[14]	8192 Hz
System clock / 8	Ck[13]	4096 Hz
System clock/ 16	Ck[12]	2048 Hz
System clock / 32	Ck[11]	1024 Hz
System clock / 64	Ck[10]	512 Hz
System clock / 128	ck [9]	256 Hz

Function	Name	32 KHz Xtal
System clock / 256	Ck[8]	128 Hz
System clock / 512	Ck[7]	64 Hz
System clock / 1024	Ck[6]	32 Hz
System clock / 2048	Ck[5]	16 Hz
System clock / 4096	Ck[4]	8 Hz
System clock / 8192	Ck[3]	4 Hz
System clock / 16384	Ck[2]	2 Hz
System clock / 32768	Ck[1]	1 Hz

Table 6.2.2 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[14] down to Ck[2], sets Ck[1]. 0 -> No action. The Read value is always '0'
1	PrIntSel	0	R/W	Interrupt select. 0 -> Interrupt from Ck[4] 1 -> Interrupt from Ck[6]
0	DebSel	0	R/W	Debouncer clock select. 0 -> Debouncer with Ck[8] 1 -> Debouncer with Ck[11] or Ck[14]

Figure 11. Prescaler Frequency Timing



With DebSel = 1 one may choose either the Ck[11] or Ck[14] debouncer frequency by selecting the corresponding metal mask option. Relative to 32kHz the corresponding max. debouncer times are then 2 ms or 0.25 ms. For the metal mask selection refer to chapter 19.1.4.
Switching the **PrIntSel** may generate an interrupt request. Avoid it with **MaskIRQ32/8** = 0 selection during the switching operation.

The prescaler contains 2 interrupt sources:
- IRQ32/8 ; this is Ck[6] or Ck[4] positive edge interrupt, the selection is depending on bit **PrIntSel.**
- IRQHz1 ; this is Ck[1] positive edge interrupt

There is no interrupt generation on reset.
The first IRQHz1 Interrupt occurs 1 sec (32kHz) after reset.

7. Input and Output ports

The EM6617 has one input port and two bi-directional ports.

7.1 Ports overview

Table 7.1.1 Input and Output Ports Overview

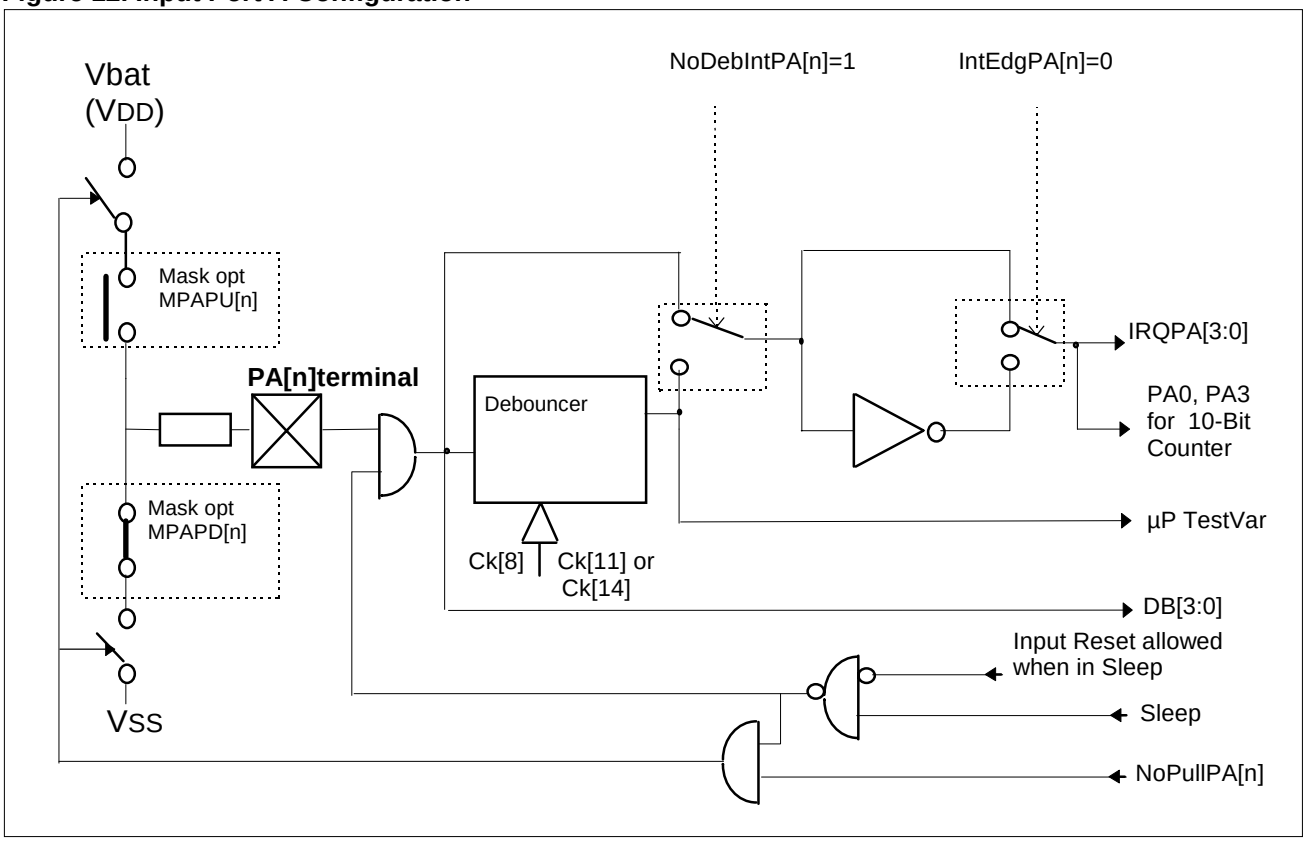
Port	Mode	Mask(M:) or Register(R:) Option	Function	Bit-wise Multifunction on Ports			
PA [3:0]	Input	M: Pull-up M: Pull-down (default) R: Pull enabling R: Debouncer or direct input for IRQ requests and Counter R: + or - for IRQ-edge and counter R: Input reset combination	-Input -Bit-wise interrupt request -Software test variable conditional jump -PA[3],PA[0] input for the event counter -Port A reset inputs	**PA[3]** 10 bit event counter clock -	**PA[2]** - -	**PA[1]** - TestVar2	**PA[0]** 10 bit event counter clock TestVar1
PB [3:0]	Individual input or output	R: CMOS or Nch. open drain output R: Pull-down on input R: Pull-up on input M: Pull-up M: Pull-down	-Input or output -PB[3] for the PWM output -PB[2:0] for the Ck[11,16,12] output -Tristate output	**PB[3]** PWM output	**PB[2]** Ck[11] output	**PB[1]** Ck[16] output	**PB[0]** Ck[12] output
PC [3:0]	Port-wise input / output	R: CMOS or Nch. open drain output R: Pull-down on input R: Pull-up on input M: Pull-up M: Pull-down	-Input or output -Tristate output	**PC[3]** only in 28 pin package	**PC[2]** only in 28 pin package	**PC[1]**	**PC[0]**

7.2 Port A

The EM6617 has one four bit general purpose CMOS input port. The port A input can be read at any time, internal pull-up or pull-down resistors can be chosen. All selections concerning port A are bit-wise 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 port A pull-up or pull-down resistors are turned off, and the inputs are deactivated except if the **InpResSleep** bit in the option register **OPTFSelPB** is set to 1. In this case the port A inputs are continuously monitored to match the input reset condition which will immediately wake the EM6617 from sleep mode (all pull resistors remain).

Figure 12. Input Port A Configuration



7.2.1 IRQ on Port A

For interrupt request generation (IRQ) one can choose direct or debouncer 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 (**RegPresc**). This means a worst case of 16 ms (default) or 2 ms (0.25 ms by metal mask) with a system clock of 32 KHz.
Either a positive or a negative edge on the port A inputs - after debouncer or not - can generate an interrupt request. This selection is done in the option register **OPTIntEdgPA.**

All four bits of port A 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 allows 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 13.
It is recommended to mask the port A IRQ's while one changes the selected IRQ edge. Else one may generate a IRQ (Software IRQ). I.e. PA[0] on '0' then changing from positive to negative edge selection on PA[0] will immediately trigger an IRQPA[0] if the IRQ was not masked.

7.2.2 Pull-up or Pull-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 the port A metal mask chapter for details. The pull resistor can be inhibited using the **NoPullPA[n]** bits in the register **OptNoPullPA.**

Table 7.2.1. Pull-up or Pull-down Resistor on Port A Inputs

Option mask pull-up MPAPU[n]	Option 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 pull-up or pull-down may be chosen on any port A terminal (one choice is excluding the other)

7.2.3 Software Test Variables

The port A terminals PA[2:0] are also used as input conditions for conditional software branches. Independent of the **OPTDebIntPA** and the **OPTIntEdgPA.** These CPU inputs always have a debouncer.

- Debounced PA[0] is connected to CPU TestVar1.
- Debounced PA[1] is connected to CPU TestVar2.
- *SWB signal SWBEmpty is connected to CPU TestVar3*

7.2.4 Port A 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 debouncer or direct input with the register **OPTDebIntPA** and non-inverted or inverted input with the register **OPTIntEdgPA**. Debouncer input is always recommended.

7.3 Port A registers

Table 7.3.1 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 port A terminal

Table 7.3.2 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 7.3.3 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 7.3.4 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 7.3.5 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 7.3.6 Register OPTNoPullPA

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

Default "0" is: depending on mask selection

7.4 Port B

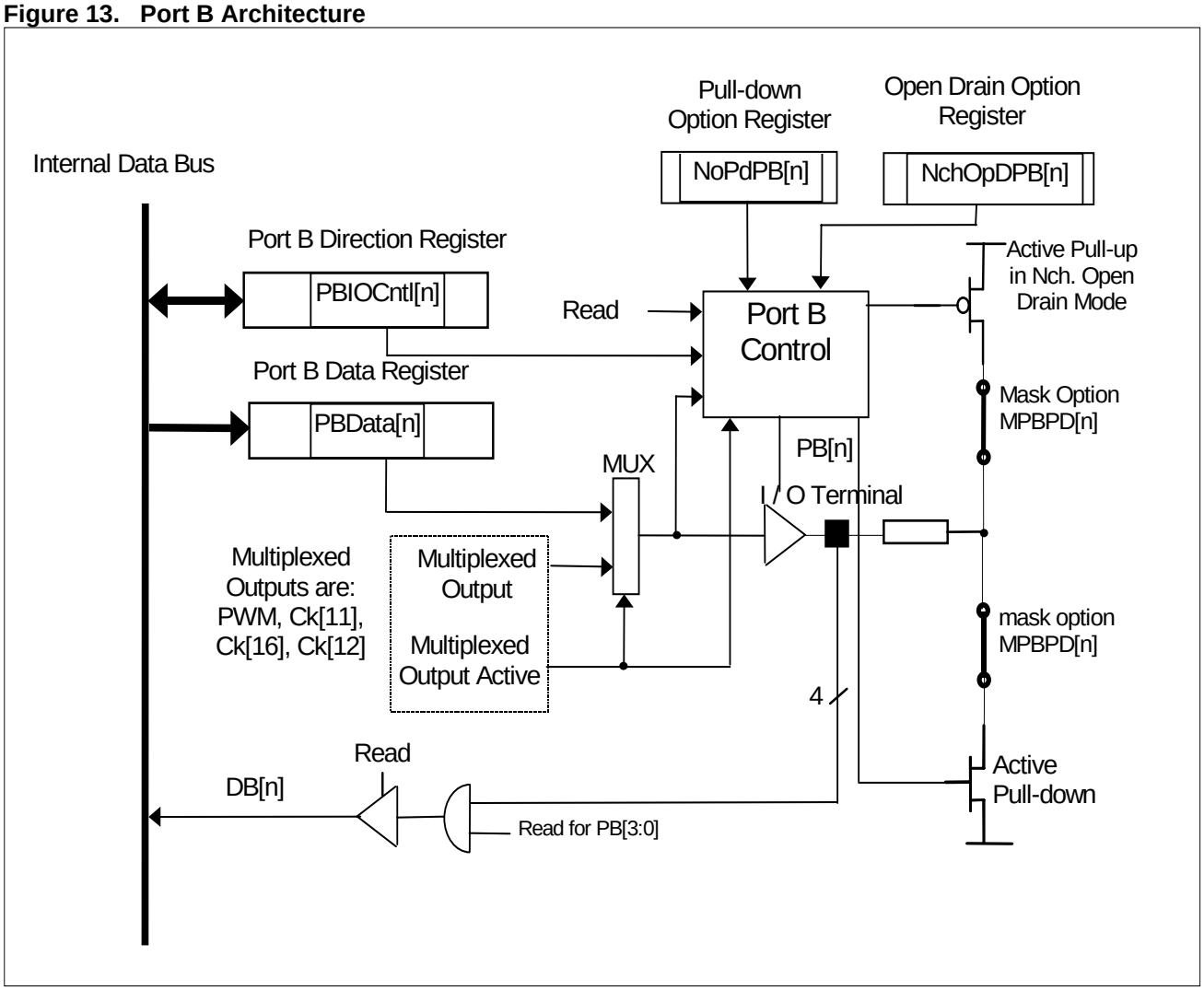
The EM6617 has one four bit general purpose I/O port. Each bit can be configured individually by software for input/output, pull-up, pull-down and CMOS or Nch. open drain output type. The port outputs either data, frequency or PWM signals.

7.4.1 Input / Output Mode

Each port B terminal is bit-wise bi-directional. The input or output mode on each port B terminal is set by writing the corresponding bit in the **RegPBCntl** control register. 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 port B terminal status can be read on address **RegPBData** even in output mode. Be aware that the data read on port B is not necessary of the same value as the data stored on **RegPBData** register.
See also Figure 13 for details.

Figure 13. Port B Architecture



7.4.2 Pull-up or Pull-down

On each terminal of PB[3:0] an internal input pull-up (metal mask MPBPU[n]) or pull-down (metal mask MPBPD[n]) resistor can be connected per metal mask option. Per default the two resistors are in place. In this case one can chose per software to have either a pull-up, a pull-down or no resistor. See below.

For Metal mask selection and available resistor values refer to 19.1.2.

Pull-down ON : MPBPD[n] must be in place ,
AND bit **NoPdPB**[n] must be '0' .

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

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

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

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

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

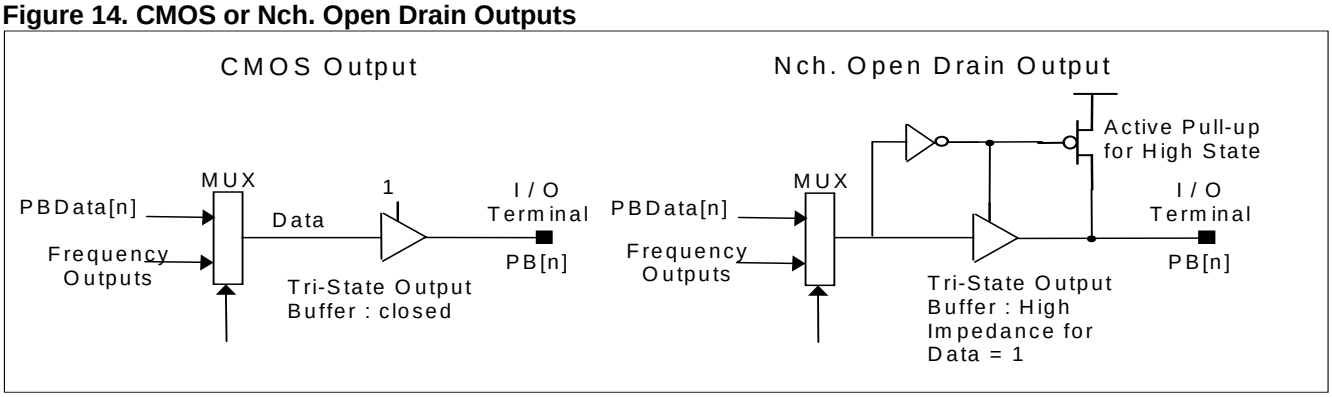
- Switch on the pull resistor.
- Allow sufficient time - RC constant - for the pull resistor to drive the line to either V_{SS} or V_{DD}.
- Read the port B
- Switch off the pull resistor

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

7.4.3 CMOS or Nch. Output

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

Figure 14. CMOS or Nch. Open Drain Outputs



7.4.4 PWM and Frequency Output

PB[3] can also be used to output the PWM (Pulse Width Modulation) signal from the 10-Bit Counter, the Ck[11], Ck[16] as well as the Ck[12] prescaler frequencies.

-Selecting PWM output on PB[3] with bit **PWMOn** in register **RegPresc** and running the counter.
-Selecting Ck[11] output on PB[2] with bit **PB1kHzOut** in register **OPTFSelPB**
-Selecting Ck[16] output on PB[1] with bit **PB32kHzOut** in register **OPTFSelPB**
-Selecting Ck[12] output on PB[0] with bit **PB2kHzOut** in register **OPTFSelPB**

7.5 Port B registers

Table 7.5.1 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 7.5.2 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: Port B in input mode

Table 7.5.3 Register OPTFSelPB

Bit	Name	power on value	R/W	Description
3	PB1kHzOut	0	R/W	ck[11] output on PB[2]
2	PB32kHzOut	0	R/W	ck[16] output on PB[1]
1	PB2kHzOut	0	R/W	ck[12] output on PB[0]
0	*InpResSleep*	*0*	*R/W*	*Reset From SLEEP with Port A*

Default "0" is: No frequency output, port A Input reset can not reset the SLEEP mode.

Table 7.5.4 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 7.5.5 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]

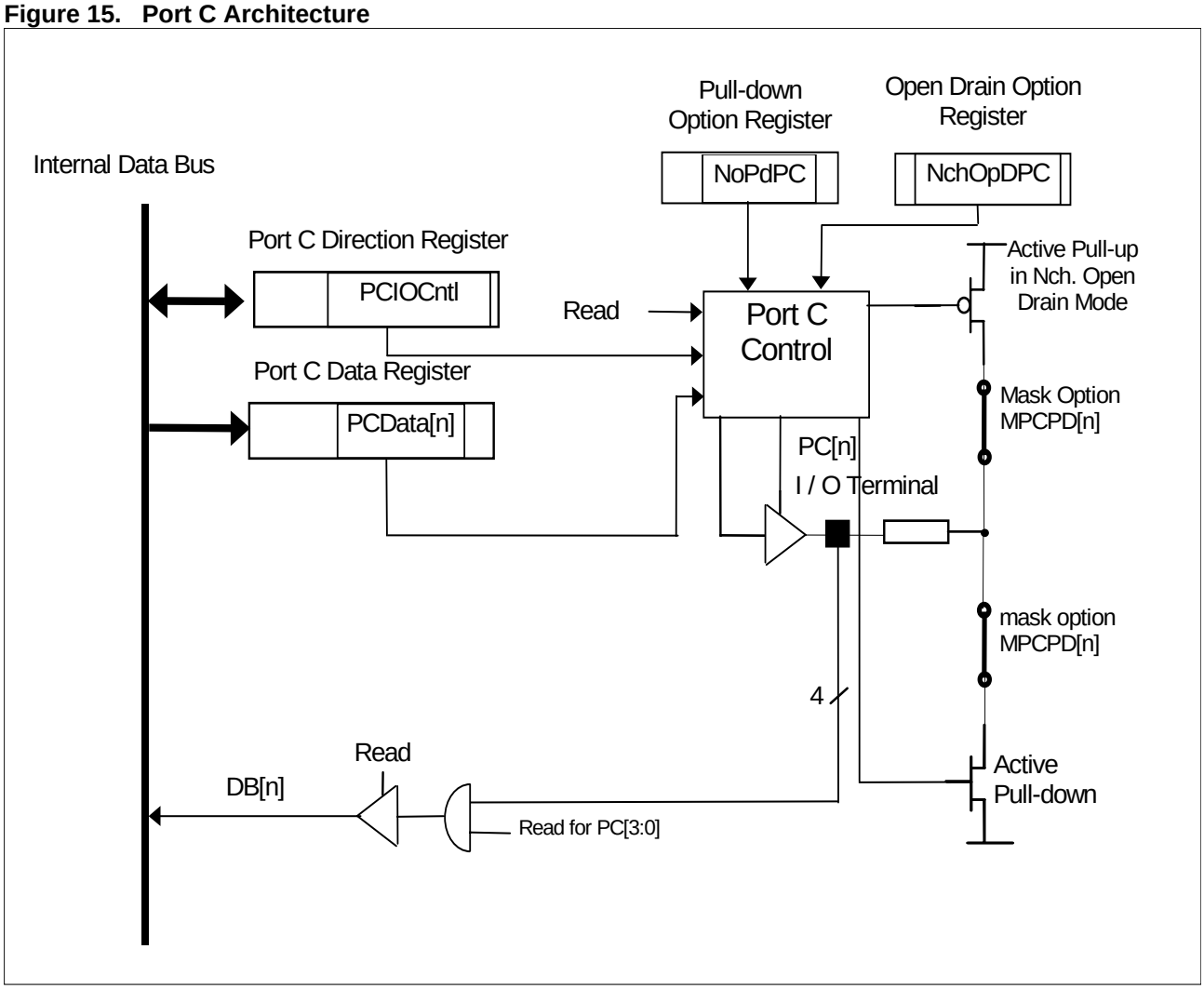
7.6 Port C

The EM6617 has one globally configurable Input / Output port which is 4 bit wide (only two bits are available for 24 pin packages). Input or output mode can be set by writing the bit **PCIOCntl** in **RegPCCntl** register.
"0" = input mode (default), "1" = output mode. The **RegPCData** register is used to write output data on port C. Input data is read directly on the input terminal and put onto the internal data bus. It is not stored in the **RegPCData** register. The port C terminal status can be read on address **RegPCData** even in output mode. Be aware that the data read on port C is not necessary of the same value as the data stored on **RegPCData** register.

At any reset, the **RegPCCntl** register is cleared, thus setting the port in input mode. During SLEEP mode, PC[3:0] are in high impedance state.

The port C is globally configurable to act as CMOS or Nch. open drain port , selectable in **OPTPCandStr** register (**NchOpDPC** bit).

Figure 15. Port C Architecture



7.6.1 Pull-up or Pull-down

On each terminal of PC[3:0] an internal input pull-up (metal mask MPCPU[n]) or pull-down (metal mask MPCPD[n]) resistor can be connected per metal mask option. Per default the two resistors are in place. In this case 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 19.1.3.

Pull-down ON : MPCPD[n] must be in place ,
AND bit **NoPdPC**[n] must be '0' .
Pull-down OFF: MPCPD[n] is not in place,
OR if MPCPD[n] is in place **NoPdPC**[n] = '1' cuts off the pull-down.
OR selecting **NchOpDPC**[n] = '1' cuts off the pull-down.
Pull-up ON : MPCPU[n] must be in place,
AND bit **NchOpDPC**[n] must be '1' ,
AND (bit **PCIOCntl =** '0' (input mode) **OR** if **PBIOCntl** = '1' while **PCData**[n] = 1.)
Pull-up OFF : MPCPU[n] is not in place,
OR if MPCPU[n] is in place **NchOpDPC**[n] = '0' cuts off the pull-up,
OR if MPCPU[n] is in place and if **NchOpDPC**[n] = '1' then **PCData**[n] = 0 cuts the pull-up.

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

For **POWER SAVING** one can switch off the port C pull resistors between two read phases. No cross current flows in the input amplifier while the port C 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 V_{SS} or V_{DD}.
- Read the port C
- Switch off the pull resistor

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

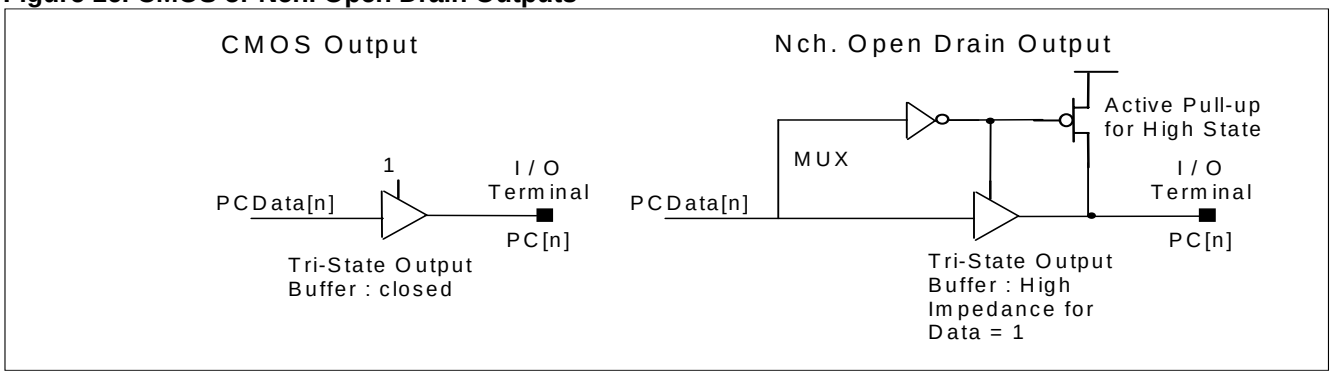
7.6.2 CMOS or Nch. Output

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

In CMOS output mode the pad can be driven high or low. Pull-ups and pull-downs are not active.
In CMOS input mode, if the corresponding metal option is in place (default), one can choose to have an internal pull-down resistor by setting to "1" the bit **NoPdPC** in **OPTPCandStr** register (default pull-down).
In N-Channel open drain mode, if the corresponding metal option is in place (default), one always has the pull-up resistor active except if the port is in output mode and drives a "0" (RegPCCntl= "1", RegPBData[n]= "0") The pull-down resistor is always off in Nch. open drain mode.

Pull-downs in CMOS input mode and weak pull-ups in Nch. open drain mode are port-wise configurable with the register settings. The metal mask options to selectively connect or disconnect pull-up or pull-down resistors can be different for each port C terminal.

Figure 16. CMOS or Nch. Open Drain Outputs



7.7 Port C Registers

Table 7.7.1 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 port C terminal (not the internal register read).

Table 7.7.2 Register RegPCCntl

Bit	Name	Reset	R/W	Description
3	--	0	R	Always reads 0
2	--	0	R	Always reads 0
1	--	0	R	Always reads 0
0	PCIOCntl	0	R/W	I/O control for port C

Default "0" is : Port C in input mode

Table 7.7.3 Option Register OPTPCandStr

Bit	Name	Power on value	R/W	Description
3	NoPdPC	0	R/W	No pull-down on port C
2	NchOpDPC	0	R/W	N-channel Open Drain port C
1	*StrobeOutSel1*	*0*	*R/W*	*Strobe output selection*
0	StrobeOutSel0	0	R/W	Strobe output selection

Default "0" is : Pull-down on, CMOS on PC[3:0]

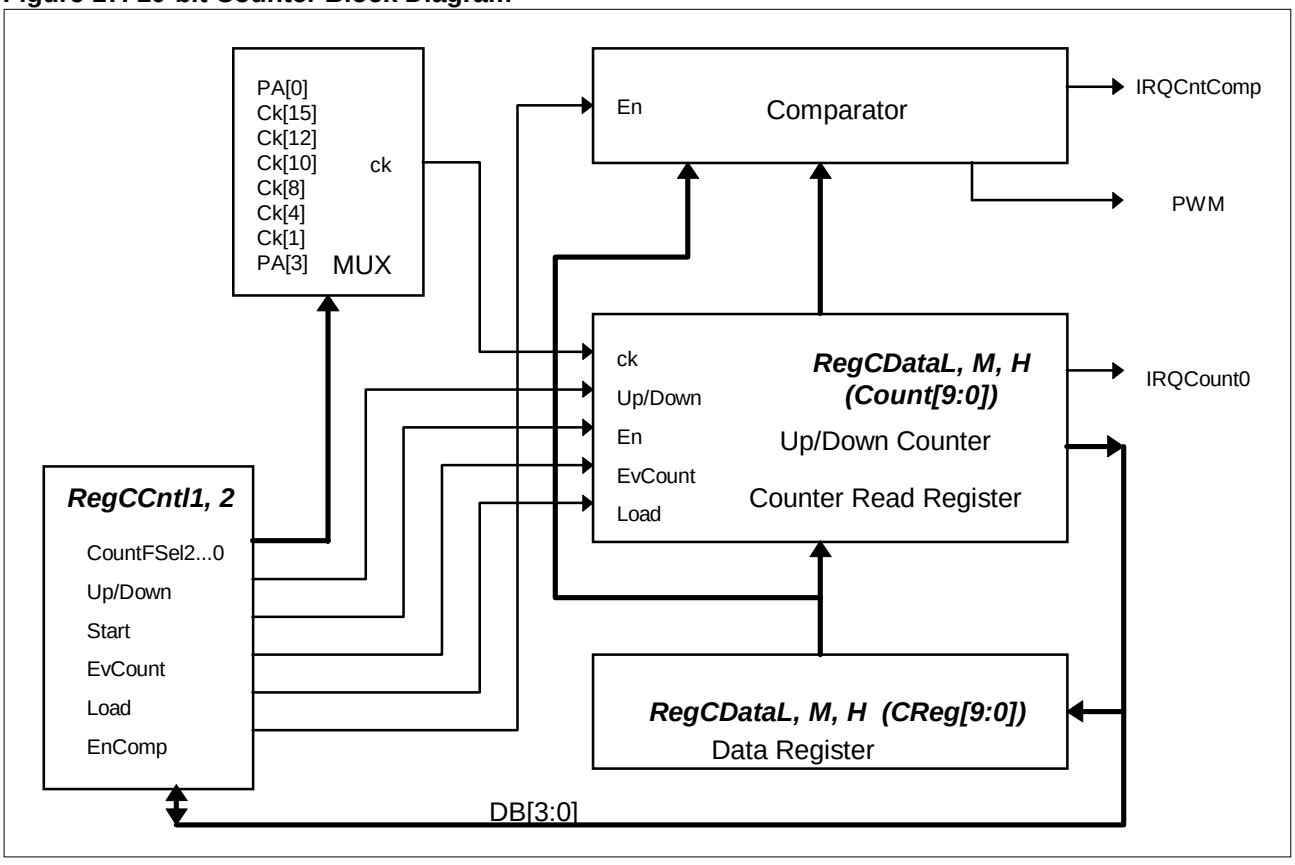
8. 10-bit Counter

The EM6617 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 derived prescaler frequencies and two 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 matches the compare data register value. Each of this interrupt requests can be masked (default). See section 13 for more information about the interrupt handling.

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

A Pulse-Width-Modulation signal (PWM) can be generated and output on port B terminal PB[3].

Figure 17. 10-bit Counter Block Diagram



8.1 Full and Limited Bit Counting

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

Table 7.7.1. Counter length selection

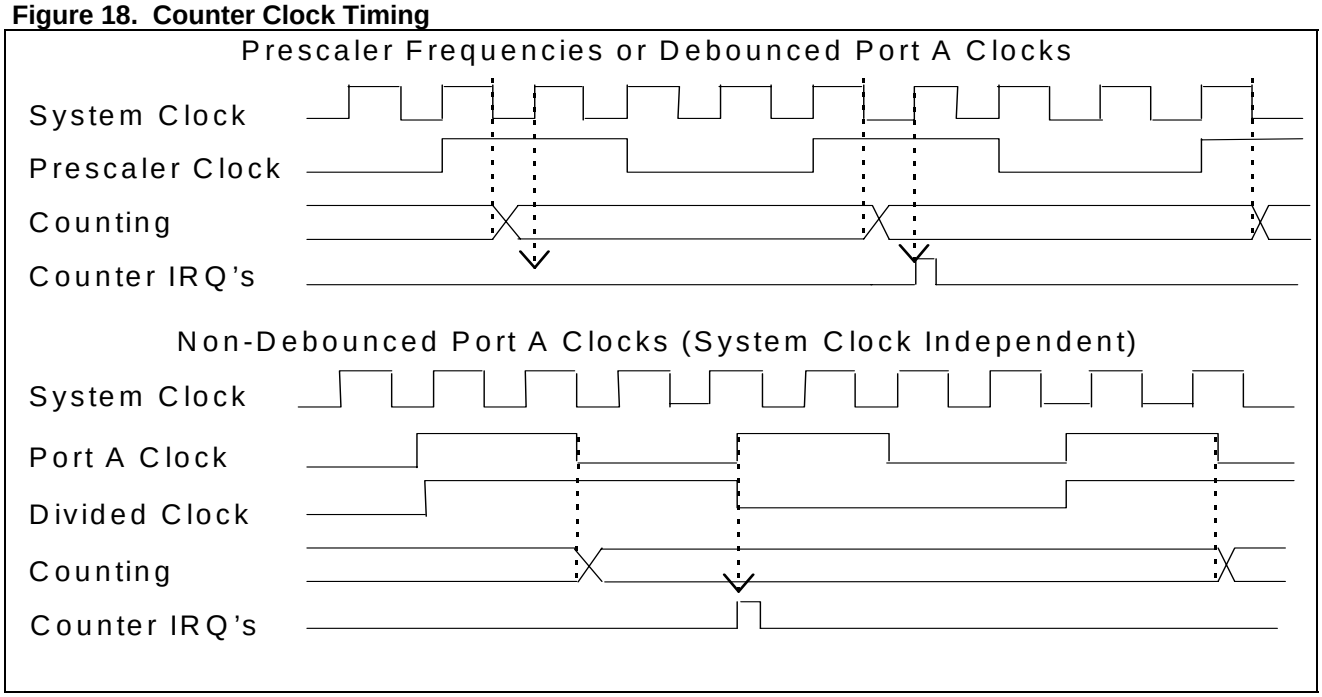
BitSel[1]	BitSel[0]	counter length
0	0	10-Bit
0	1	8-Bit
1	0	6-Bit
1	1	4-Bit

8.2 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 17.07 minutes (1Hz clock, 10 bit length) or as little as 977 µs (Ck[15], 4 bit length). The **IRQCount0**, generated at each roll over, can be used for time bases, measurements length definitions, input polling, wake up from Halt mode, etc. The **IRQCount0** and **IRQComp** are generated with the system clock Ck[16] rising edge. IRQCount0 condition in up count mode is : reaching 3FF if 10-bit counter length (or FF, 3F, F in 8, 6, 4-bit counter length). In down count mode the condition is reaching '0'. The non-selected bits are 'don't care'. For IRQComp refer to section 8.4.

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

Figure 18. Counter Clock Timing



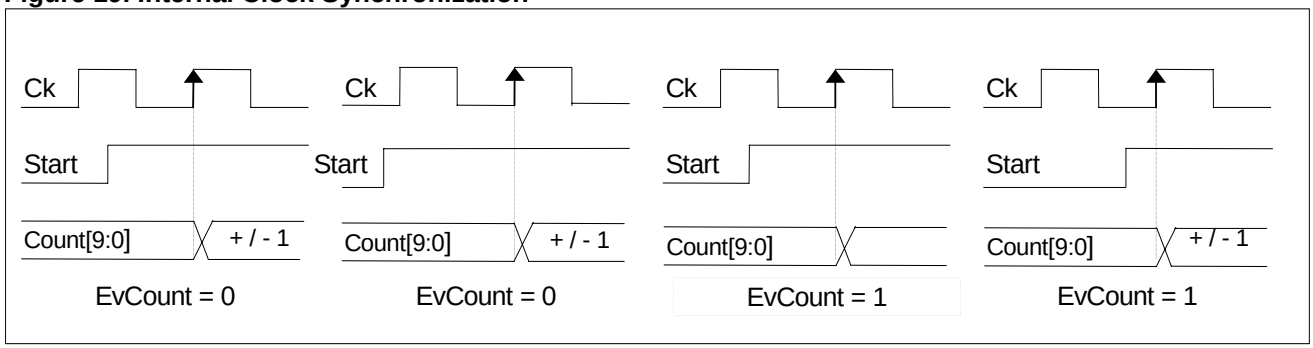
The two remaining clock sources are coming from the PA[0] or PA[3] terminals. Refer to the Figure 12 on page 14 for details. Both sources can be either debounced (Ck[11] or Ck[8]) or direct inputs, the input polarity can also be chosen. The output after the debouncer polarity selector is named PA3 , PA0 respectively. For the debouncer and input polarity selection refer to chapter 7.2.4.

In the case of port A input clock without debouncer, the counting clock frequency will be half the input clock on port A. The counter advances on every odd numbered port A negative edge (divided clock is high level). IRQCount0 and IRQComp will be generated on the rising PA3 or PA0 input clock edge. In this condition the EM6617 is able to count with a higher clock rate as the internal system clock (Hi-Frequency Input). Maximum port A input frequency is limited to 200kHz. If higher frequencies are needed, please contact EM-Marin.
In both, up or down count (default) mode, the counter is cyclic. The counting direction is chosen in register **RegCCntl1** bit **Up/Down** (default '0' is down count). The counter increases or decreases its value with each positive clock edge of the selected input clock source. Start up synchronization is necessary because one can not always know the clock status when enabling the counter. With EvCount=0, the counter will only start on the next positive clock edge after a previously latched negative edge, while the **Start** bit was already set to '1'. This synchronization is done differently if event count mode (bit **EvCount**) is chosen. Refer also to Figure 19. Internal Clock Synchronization.

8.3 Event Counting

The counter can be used in a special event count mode where a certain number of events (clocks) on the PA[0] or PA[3] input are counted. In this mode the counting will start directly on the next active clock edge on the selected port A input.
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** and should be debounced. The debouncer is switched on in register **OPTDebIntPA** bits NoDebIntPA[3,0]=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. Refer also to Figure 12 for internal clock signal generation.

Figure 19. Internal Clock Synchronization



8.4 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 an interrupt (**IRQComp**) is generated. The compare function is switched on with the bit **EnComp** in the register **RegCCntl2**. With **EnComp** = 0 no **IRQComp** is generated. Starting the counter with the same value as the compare register is possible, no IRQ is generated on start. Full or Limited bit compare are possible, defined by bit **SelIntFull** in register **RegSysCntl1**.
EnComp must be written after a load operation (**Load** = 1). Every load operation resets the bit EnComp.

Full bit compare function.
Bit **SelIntFull** is set to '1'. The function behaves as described above independent of the selected counter length. Limited bit counting together with full bit compare can be used to generate a certain amount of IRQCount0 interrupts until the counter generates the IRQComp interrupt. With **PWMOn**='1' the counter would have automatically stopped after the IRQComp, with **PWMOn**='0' it will continue until the software stops it. **EnComp** must be cleared before setting SelIntFull and before starting the counter again. Be careful, PWMOn also redefines the port B PB[3] output data.(refer to section 8.5).

Limited bit compare
With the bit **SelIntFull** set to '0' (default) the compare function will only take as many bits into account as defined by the counter length selection **BitSel[1:0]** (see chapter 8.1).

8.5 Pulse Width Modulation (PWM)

The PWM generator uses the behavior of the Compare function (see above) so **EnComp** must be set to activate the PWM function.. At each Roll Over or Compare Match the PWM state - which is output on port B PB[3] - will toggle. The start value on PB[3] is forced while **EnComp** is 0 the value is depending on the up or down count mode. Every counter value load operation resets the bit **EnComp** and therefore the PWM start value is reinstalled.
Setting **PWMOn** to '1' in register **RegPresc** routes the counter PWM output to port B terminal PB[3]. Insure that PB[3] is set to output mode . Refer to section 7.4 for the port B setup.
The PWM signal generation is independent of the limited or full bit compare selection bit **SelIntFull.** However if **SelIntFull** = 1 (FULL) and the counter compare function is limited to lower than 10 bits one can generate a predefined number of output pulses. In this case, the number of output pulses is defined by the value of the unused counter bits. It will count from the start value until the IRQComp match.
One must not use a compare value of hex 0 in up count mode nor a value of hex 3FF (or FF,3F, F if limited bit compare) in down count mode.

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

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

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

8.5.1 How the PWM Generator works.

For Up Count Mode; Setting the counter in up count and PWM mode the PB[3] PWM output is defined to be 0 (**EnComp**=0 forces the PWM output to 0 in upcount mode, 1 in downcount). Each Roll Over will set the output to '1' and each Compare Match will set it back to '0'. The Compare Match for PWM always only works on the defined counter length. This, independent of the SelIntFull setting which is valid only for the IRQ generation. Refer also to the compare setup in chapter 8.4.

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

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

Figure 20. PWM Output in Up Count Mode

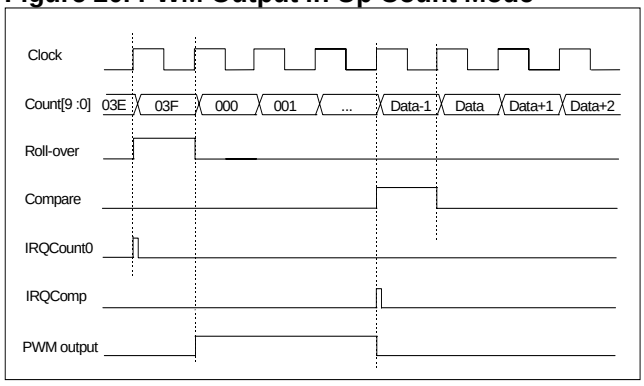
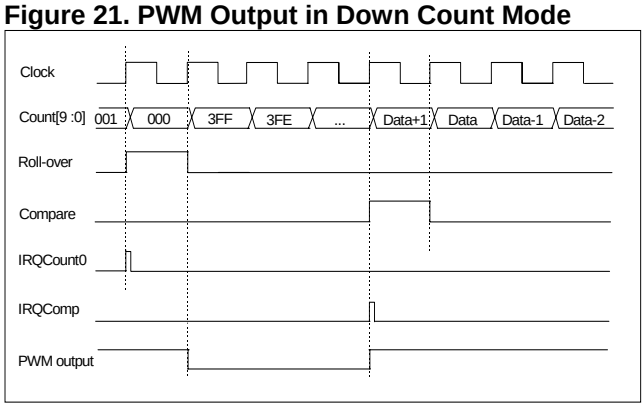


Figure 21. PWM Output in Down Count Mode



In Down Count Mode everything is inverted. The PWM output starts with the '1' value. Each Roll Over will set the output to '0' and each Compare Match will set it back to '1'. For limited pulse generation one must load the complementary pulse number value. I.e. for 5 pulses counting on 4 bits load bits[9 :4] with hex 3A (bin 111010).

8.5.2 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[15]	-> 977 µs	(32 KHz)
the maximum signal period is	: 1024 x Fmin*	-> 1024 x 1/Ck[1]	-> 1024 s	(32 KHz)
the minimal pulse width is	: 1 bit	-> 1 x 1/Ck[15]	-> 61 µs	(32 KHz)

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

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

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

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

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

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

How to use the counter;

If PWM output is required one has to put the port B[3] in output mode and set PWMOn=1 in step 5.

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

8.7 10-bit Counter Registers

Table 8.7.1 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 8.7.2 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[15]
0	1	0	Prescaler Ck[12]
0	1	1	Prescaler Ck[10]
1	0	0	Prescaler Ck[8]
1	0	1	Prescaler Ck[4]
1	1	0	Prescaler Ck[1]
1	1	1	Port A PA[3]

Table 8.7.3 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 8.7.4 Register RegSysCntl1

Bit	Name	Reset	R/W	Description
3	*IntEn*	*0*	*R/W*	*General interrupt enable*
2	*SLEEP*	*0*	*R/W*	*Sleep mode*
1	SelIntFull	0	R/W	Compare Interrupt select
0	*ChTmDis*	*0*	*R/W*	*For EM test only*

Default : Interrupt on limited bit compare

Table 8.7.5 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 8.7.6 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 8.7.7 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

Table 8.7.8 Counter Length Selection

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

9. Serial (Output) Write Buffer - SWB

The EM6617 has simple Serial Write Buffer which outputs serial data and serial clock.

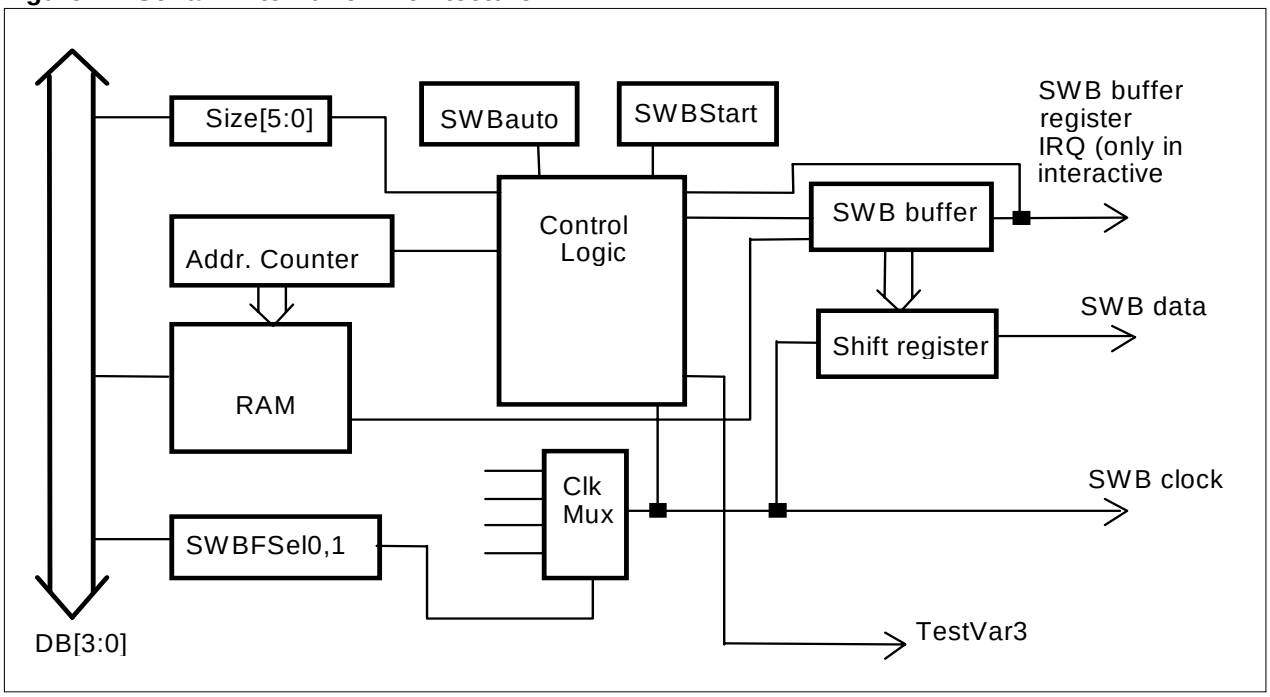
Serial Write Buffer clock frequency is selected by bits **SWBFSel0** and **SWBFSel1** in **RegSWBCntl** register. The possible values are 1kHz (default), 2kHz, 8kHz or 16kHz.

The signal TestVar[3], which is used by the processor to make conditional jumps, indicates "Transmission finished" in automatic send mode or "SWBbuffer empty" in interactive send mode. In interactive mode, TestVar[3] is equivalent to the interrupt request flags stored in **RegIRQ[i]** registers : it permits to recognize the interrupt source. (See also the interrupt handling section 13 for further information). To serve the "SWBbuffer empty " interrupt request, one only has to make a conditional jump on TestVar[3].

The SWB data is output on the rising edge of the clock. Consequently, on the receiver side the serial data can be evaluated on falling edge of the serial clock edge.

Normally the Clock and the Data output terminals are always driven to '0' outside a SWB data transfer. With a metal option one can put the Data output, the Clock output or both into a high impedance state outside of a SWB transfer. Refer to 19.1.4 for the mask settings. The timing going into high impedance state into SWB transfer and back into high impedance is depending on the selected mode, interactive or automatic.

Figure 22. Serial Write Buffer Architecture



SWB has two operational modes, automatic mode and interactive mode.

9.1 SWB Automatic send mode

Automatic mode enables a buffer on a predefined length to be sent at high transmission speeds up to ck[15] (16kHz). In this mode user prepares all the data to be sent (minimum 8 bits, maximum 256 bits) in the RAM. The user then selects the clock speed, sets the number of data nibbles to be sent, selects automatic transmission mode (**SWBAuto** bit set to 1) and enters STANDBY mode by executing a HALT instruction. Once the HALT instruction is activated the SWB peripheral module sends the data in register **RegSWBuff** followed by the data in the RAM starting at address 00 up to the address specified by the bits **size[5:0]** located in the **RegSWBSizeL, RegSWBSizeH** registers.

During automatic transmission the general INTEN bit is disabled automatically to prevent other Interrupts to reset the standby mode. At the end of automatic transmission EM6617 leaves standby mode *(**INTEN** is automatically Enabled)* and sets TestVar[3] high. TestVar[3] = 1 is signaling SWB transmission is terminated.

As soon as SWBAuto is high, the general IntEn flag is disabled until the SWBAuto goes back low.

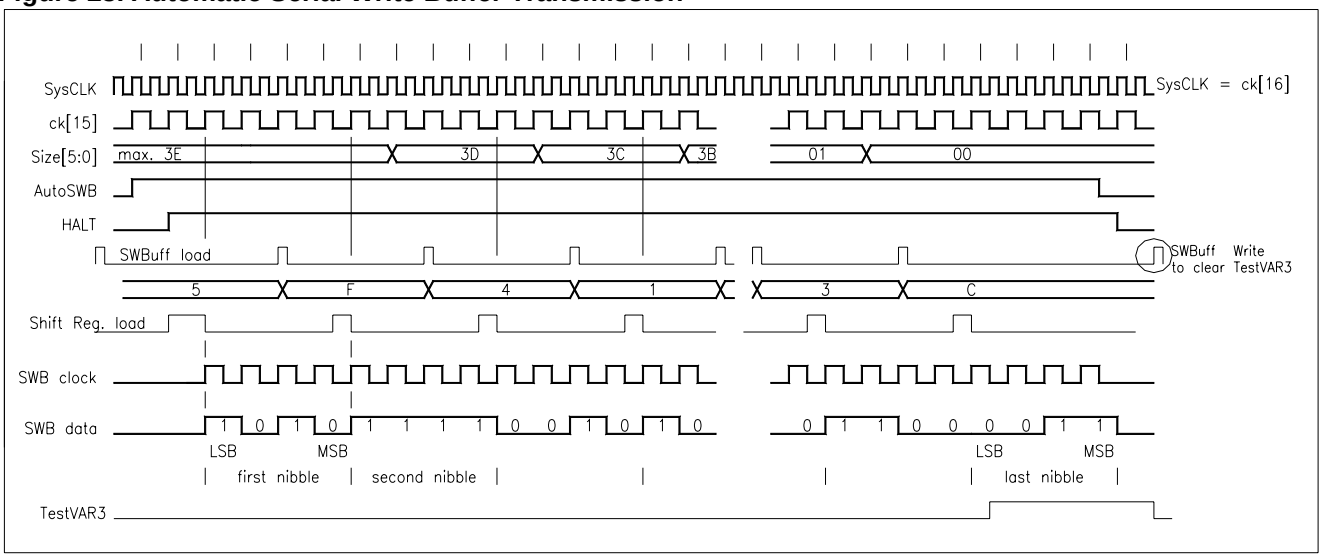
*After automatic SWB transmission **INTEN** bit becomes high. Although set to 1 via the Halt instruction the bit **INTEN** is disabled throughout the whole SWB automatic transmission. It resumes to 1 at the end of transmission*.

The data to be sent must be prepared in the following order:
First nibble to be sent must be written in the **RegSWBuff** register . The other nibbles must be loaded in the RAM from address 00 (second nibble at adr.00, third at adr.01,...) up to the address with last nibble of data to be send = "**size**" address. Max. address space for SWB is 3E ("**size**" 3E hex) what gives together with **RegSWBuff** up to 64 nibbles (256 bits) of data to be sent. The minimum amount of data bits one can send in automatic SWB mode is 8 . In this case the last RAM address to be sent is 00 ("size" = 00).

Once data are written into the RAM and into the RegSWBuff, the user has to load the "**size**" (adr. of the last nibble to be send - bits size[5 :0]) into the **RegSWBSizeL** and **RegSWBSizeH** register, later register together with SWBAuto =1 bit.
Now everything is ready for automatic serial transmission. To start the transmission one has to put the EM6617 in standby mode with the HALT instruction. When transmission is finished TESTvar[3] (can be used for conditional jumps) becomes active High, the bit **SWBAuto** is cleared , the processor is leaving the Standby mode and **IntEn** is switched on.
The processor now starts to execute the first instruction placed after the HALT instruction (for instance write of RegSWBuff register to clear TESTvar[3]), except if there was a IRQ during the serial transmission. In this case the CPU will go directly in the interrupt routine.

TestVar[3] stays high until **RegSWBuff** is rewritten. Before starting a second SWB action this bit must be cleared by performing a dummy write on **RegSWBuff** address.

Figure 23. Automatic Serial Write Buffer Transmission



Because the data in the RAM are still present one can start transmitting the same data once again only by recharging the RegSWBuff , RegSWBSizeL and RegSWBSizeH register together with SWBAuto bit and putting the EM6617 in HALT mode. This will start a new transmission.

Using the SWB high impedance mask option in automatic mode. As soon as one goes into Halt mode the SWB outputs go to '0' and SWB transfer starts. At the end of the transfer the SWB outputs go immediately back into high impedance state.

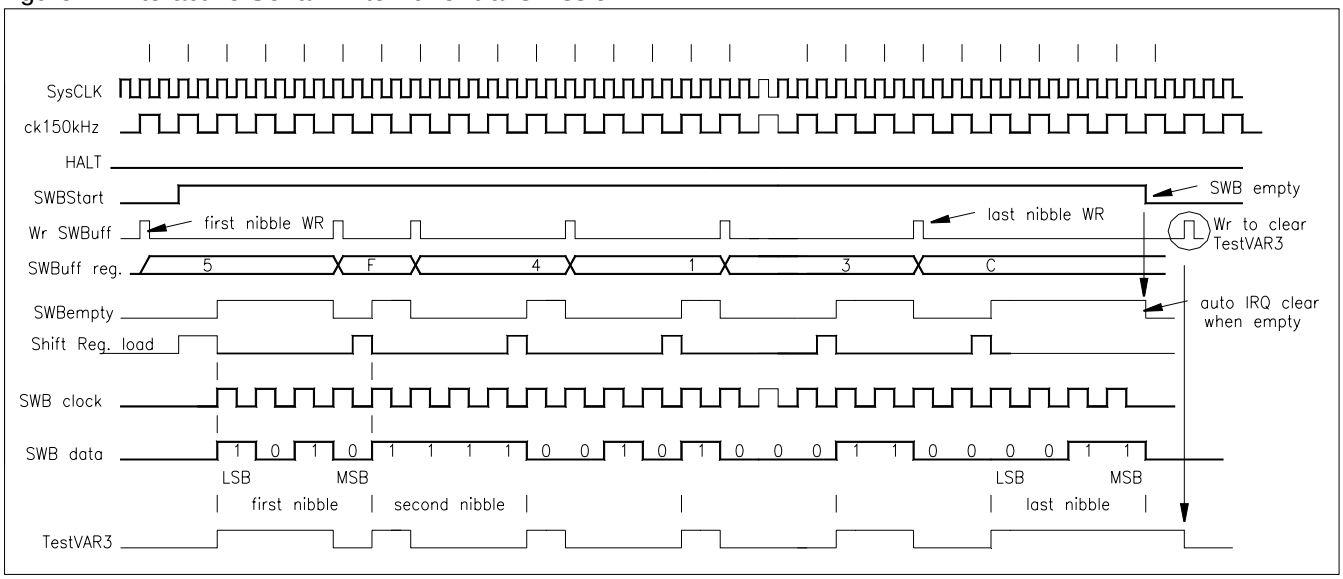
9.2 SWB Interactive send mode

In interactive SWB mode the reloading of the data transmission register **RegSWBuff** is performed by the application program. This means that it is possible to have an unlimited length transmission data stream. However, since the application program is responsible for reloading the data a continuous data stream can only be achieved at Ck[11] or Ck[12] (1 KHz or 2 KHz) transmission speeds. For the higher transmission speeds a series of writes must be programmed and the serial output clock will not be continuous.
Serial transmission using the interactive mode is detailed in Figure 24. Programming of the SWB in interactive is achieved in the following manner:

Select the transmission clock speed using the bits **SWBFSel1** and **SWBFSel0** in the **RegSWBCntl** register.
Load the first nibble of data into the SWB data register **RegSWBuff**
Start serial transmission by selecting the bit **SWBStart** in the register **RegSWBSizeH** register.

Once the data has been transferred into the serial transmission register a non maskable interrupt (SWBEmpty) is generated and TestVar[3] goes high. The CPU goes in the interrupt routine, with the JPV3 as first instruction in the routine one can immediately jump to the SWB update routine to load the next nibble to be transmitted into the **RegSWBuff** register. If this reload is performed before all the serial data is shifted out then the next nibble is automatically transmitted. This is only possible at the transmission speeds of Ck[11] or Ck[12] due to the number of instructions required to reload the register. At the higher transmission speeds of Ck[14] or Ck[15] (8 KHz or 16 KHz) the application must restart the serial transmission by writing the **SWBStart** in the **RegSWBSizeH** register after writing the next nibble to the **RegSWBuff** register.
Each time the **RegSWBuff** register is written the "SWBbuffer empty interrupt" and TestVar[3] are cleared to "0". For proper operation the **RegSWBuff** register must be written before the serial clock drops to low during sending the last bit (MSB) of the previous data.

Figure 24 Interactive Serial Write Buffer transmission



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. Be careful because if **SWBStar**t bit is cleared by software, transmission is stopped immediately.

Using the SWB high impedance mask option in Interactive mode. As soon as one sets the start bit the SWB outputs go to ‘0’ and SWB transfer starts. At the end of the transfer the SWB outputs go immediately back into high impedance state.

9.3 SWB registers

Table 9.3.1 SWB clock selection register RegSWBCntl

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

Table 9.3.2 Serial Write Buffer clock selection

SWB clock output	SWBFSel1	SWBFSel0
1024 Hz	0	0
2048 Hz	0	1
8192 Hz	1	0
16384 Hz	1	1

Table 9.3.3 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 9.3.4 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 9.3.5 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

10. 2-Channel ADC (8-bit digital converter)

The EM6617 contains one 8-bit ADC with 2 independent input channels. In addition one can choose also Vref and Vgnd as ADC input. The control logic uses an internal analogue multiplexor to select the channel to be converted. When data conversion is complete, indicated by a DATA VALID signal, the control logic saves the converted value in the **RegADCDataH** and the Shadow Low registers. **RegADCDataL** registers is updated when reading the **RegADCDataH** register. At the end of conversion an interrupt **IRQADC** is sent to the µP. The architecture of the ADC is illustrated in Figure 25.

Figure 25. ADC 8 Bit Architecture

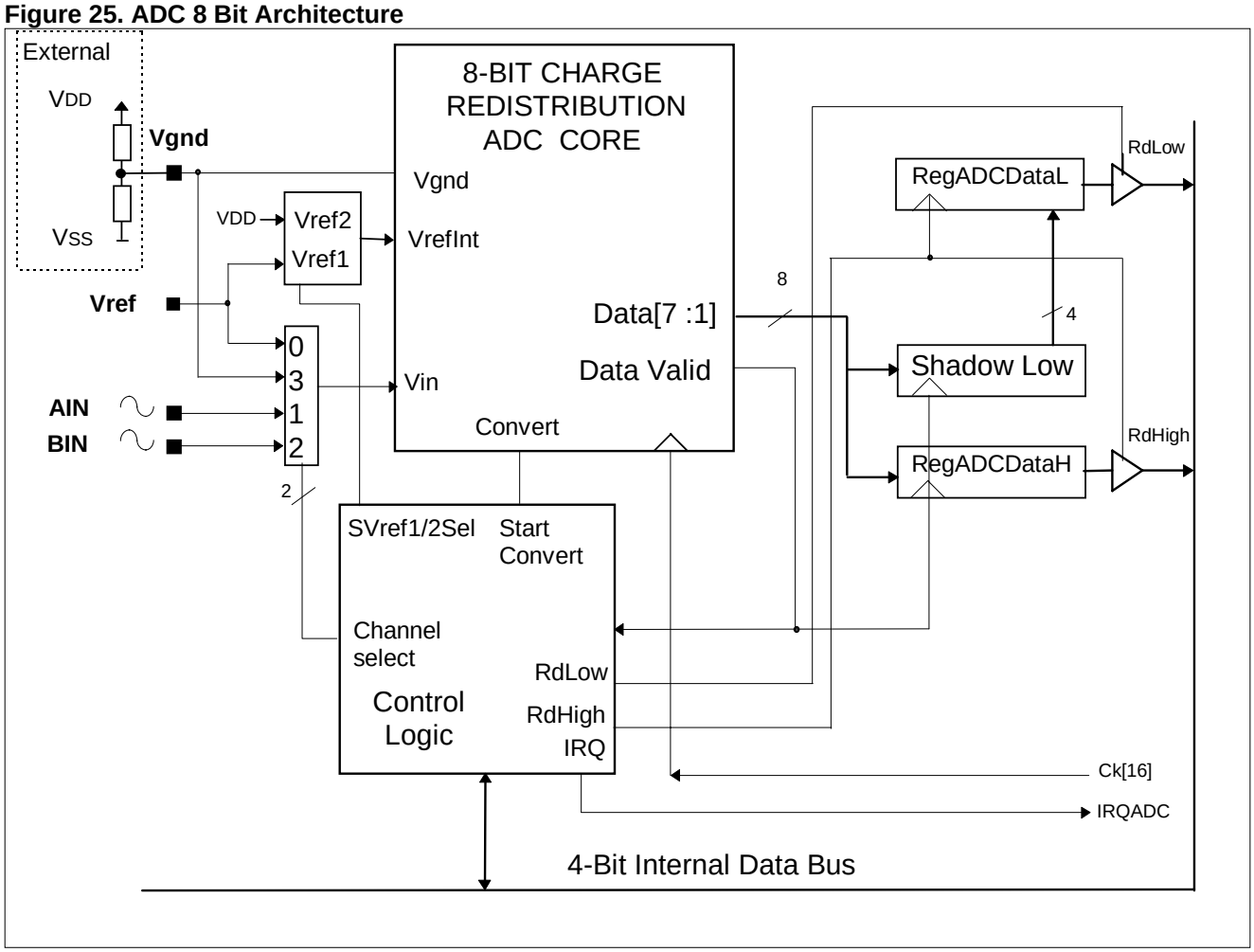
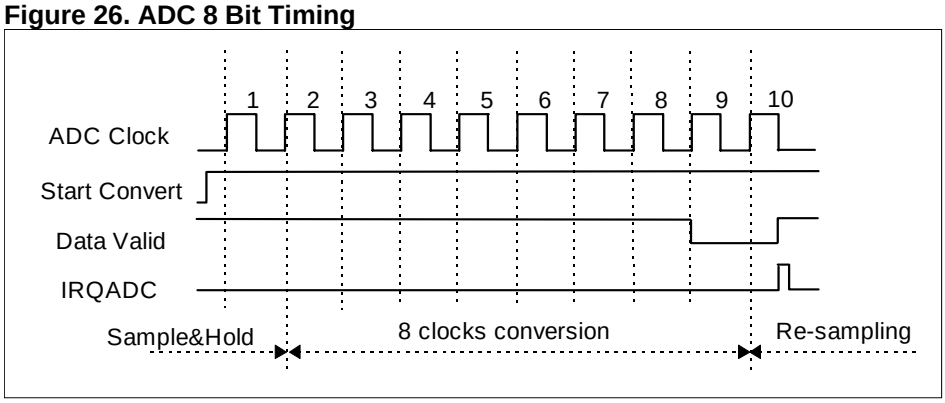


Figure 26. ADC 8 Bit Timing



The ADC 8 bit contains an inherent sample and hold function : the input voltage is sampled during acquisition phase (2 clock cycles) and is held until the end of conversion. Total conversion time is 10 clock cycles.

The ADC is of bipolar type : Positive or negative input signal referred to the virtual ground point Vgnd are converted. The virtual ground point Vgnd is ideally on (VSS + VBAT)/2 voltage level and must be supplied from external circuitry.

The positive reference voltage VREF referred to the virtual ground point Vgnd defines the input voltage range without overflow (full scale conversion : +/- Vref referred to Vgnd)

Data format is the following : MSB **ADCData[7]** is a sign bit indicating if input signal Vin is higher than virtual ground (ADCData[7] = “1“) or lower (ADCData[7] = “0“).
For negative input signal the LSB’s are coded in 1’ complement.

For instance :	Vin = +VREF	->	ADCData[7:0] = “1 1111111“	+127
	Vin = +VREF/ 2	->	ADCData[7:0] = “1 0111111“	+63
	Vin = Vgnd	->	ADCData[7:0] = “1 0000000“	+0
	Vin = -VREF/ 2	->	ADCData[7:0] = “0 1000000“	-63
	Vin = -VREF	->	ADCData[7:0] = “0 0000000“	-127

The input channel to be converted is selected by **ChannelSelA** and **ChannelSelB** bits in **RegADCCntl** register. The default channel selection is Vref as ADC input.

Setting to “1“ the **Vref1/2Sel** bit in **RegSysCntl 3** selects the internal V_{DD} (Vref2 input) as the reference voltage. By default, VREF is defined by the external Vref pad (Vref1 input).

The ADC has two working modes (continuous or single mode) selected by the **Single** bit (Single = “0“ --> continuous mode ; Single = “1“ --> single mode).

10.1 Continuous mode

The conversion process is activated by setting to “1“ the **StartConvert** bit. The selected channel is cyclically (3.2kHz) converted and the result is stored in **RegADCDataL** and **RegADCDataH** registers. When the **StartConvert** bit is set to “0“, the process runs until completion of the current 10 clock cycles and then stops. After each completion, an interrupt request **IRQADC** is generated. This interrupt request can be masked (default) (**MaskIRQADC** bit). See also the interrupt handling section 13 for further information. One always needs to read **RegADCDataH** first , this read updates the **RegADCDataL** value (shadow register).

10.2 Single mode

Setting to “1“ the **StartConvert** bit activates 1 conversion of the selected channel. At the end of the conversion, the **StartConvert** bit is automatically cleared and **IRQADC** is generated. Data are available in **RegADCDataL** and **RegADCDataH** registers. One always needs to read **RegADCDataH** first , this read updates the **RegADCDataL** value (shadow register).

10.3 2-Channel ADC registers

Table 10.3.1 ADC control register **RegADCCntl**

Bit	Name	Reset	R/W	Description
3	StartConvert	0	W	Start conversion
3	ADCBusy		R	ADC busy flag
2	Single	0	R/W	Single mode
1	ChannelSelB	0	R/W	Input channel selection
0	ChannelSelA	0	R/W	Input channel selection

Default : continuous mode, Vref pad as input channel

Table 10.3.2 Input channel selection

ChannelSelA	ChannelSelB	Input channel
0	0	Vref pad
0	1	Bin
1	0	Ain
1	1	Vgnd

Table 10.3.3 ADC data low register **RegADCDataL**

Bit	Name	Reset	R/W	Description
3	ADCdata[3]	0	R/W	ADC data bit 3
2	ADCdata[2]	0	R/W	ADC data bit 2
1	ADCdata[1]	0	R/W	ADC data bit 1
0	ADCdata[0]	0	R/W	ADC data bit 0

Table 10.3.4 ADC data high register **RegADCDataH**

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

Table 10.3.5 Control register **RegSysCntl3**

Bit	Name	Reset	R/W	Description
3	Vref1/2Sel	0		Reference voltage selection for ADC
2	--			
1	*NoOscWD*	*0*	*R/W*	*No oscillator watchdog*
0	*NoLogicWD*	*0*	*R/W*	*No logic watchdog*

Default : external Vref for the voltage reference

11. EEPROM (64 × 8 Bit)

The EEPROM addressing is indirect using 6 bits (64 addresses) defined in **RegEEPAdr** and **RegEEPCntl** registers. The EEPROM consist of 2 pages 32x8bit each, address **EEPAdr[4 :0]**. The page is selected in the **RegEEPCntl** register bit **EEPage**. So the user can address the EEPROM as it would be one block of 64x8 bit.

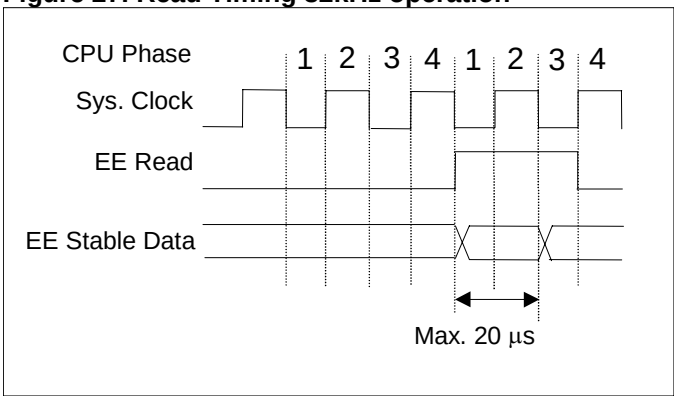
Any access to the EEPROM is done in two phases. 1st, one needs to define the address location. 2nd, one needs to start the desired action, read or write. Refer to the examples below..

How to read data from EEPROM :
1st inst. : write EEPROM address (4 low bits) in **RegEEPAddr** register.
2nd inst. : write the high address bit, page and select reading operation in **RegEEPCntl.** (**EEPAdr[4]**, **EEPage**, **EEPRdWr**=0)
3rd inst. : NOP instruction in case of 128kHz operation (metal option setting).
4th inst. : read EEPROM low data in **RegEEPDataL** register.
5th inst. : 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 inst. : write EEPROM address (4 low bits) in **RegEEPAdr** register.
2nd inst. : write EEPROM low data in **RegEEPDataL** register.
3rd inst.: write EEPROM high data in **RegEEPDataH** register.
4th inst. : write the high address bit, page and select writing operation in **RegEEPCntl**. (**EEPAdr[4]**, **EEPage**, **EEPRdWr**=1)
5th . : **IRQEEP** is generated at the end of write.
The three first instructions can be executed in any order.

Writing **RegEEPCntl** register starts automatically EEPROM reading or writing operation according to the bit **EEPRdWr**.

Figure 27. Read Timing 32kHz operation



EEPROM access time is max. 20µs : Data is available in **RegEEPDataL** and **RegEEPDataH** registers at the instruction following the read access on 32kHz system clock The read signal is 1.5 system clock wide. The CPU reads at end of phase 3.
With the 128kHz metal option the EERead signal is 3.5 system clock cycles wide. Using this option the user must use a NOP instruction before actually reading the RegEEPDataL,H values.

EEPROM writing operation lasts 24ms (Erase followed by write). The flag **EEPBusy** in **RegEEPCntl** register stays high until the writing operation is finished. An interrupt request **IRQEEP** is generated at the end of each writing operation. While **EEPBusy** is high the EEPROM must not be used at all. The EEPROM interrupt request can be masked (default) (**MaskIRQEEP** bit). See also the interrupt handling section 13 for further information .

Note : Any Reset or sleep mode will immediately cancel the EEPROM write operation. The data to be stored at this time may be corrupted.

11.1 EEPROM registers

Table 11.1.1 EEPROM control register RegEEPCntl

Bit	Name	Reset	R/W	Description
3	EEPage	0	R/W	EEPROM page select
2	EEPBusy	0	R	EEPROM writing operation busy flag
1	EEPRdWr	0	R/W	EEPROM operation read=0 / write=1
0	EEPAdr[4]	0	R/W	EEPROM`address bit 4

Writing this register starts automatically EEPROM reading or writing operation

Table 11.1.2 EEPROM address register RegEEPAdr

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

Table 11.1.3 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 11.1.4 EEPROM data high register RegEEPDataH

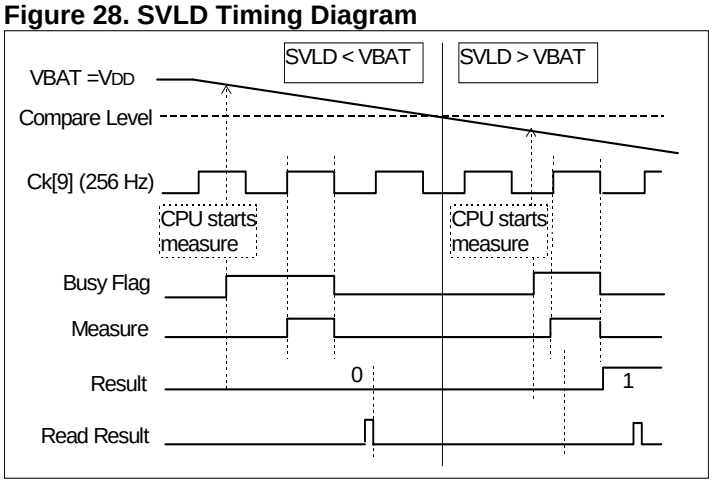
Bit	Name	Reset	R/W	Description
3	EEPdata[7]	0	R/W	EEPROM data bit 7
2	EEPdata[6]	0	R/W	EEPROM data bit 6
1	EEPdata[5]	0	R/W	EEPROM data bit 5
0	EEPdata[4]	0	R/W	EEPROM data bit 4

12. Supply Voltage Level Detector

The EM6617 has a built-in Supply Voltage Level Detector (SVLD) circuitry, such that the CPU can compare the supply voltage against a pre-selected value. 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 (2 ms at 32 KHz). The busy flag **VldBusy** stays high from **VldStart** set until the measurement is finished. The worst case time until the result is available is 1.5 Ck[9] prescaler clock periods (32 KHz -> 6 ms). The detection level must be defined in register **RegVldLevel** before the **VldStart** bit is set.
During the actual measurement (2 ms) the device will draw an additional 5 µA of I_{VDD} current. After the end of the measure the result is available by inspection of the bit **VldResult**.
An interrupt **IRQVLD** is send to indicate the end of measure. 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 guaranteed. The interrupt request can be masked (default) (**MaskIRQVLD** bit).

Figure 28. SVLD Timing Diagram



12.1 SVLD Register

Table 12.1.1 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	VLDlevel1	0	R/W	Vld level selection
0	VLDlevel0	0	R/W	Vld level selection

R*; VLDResult is not guaranteed while VLDBusy=1

Table 12.1.2 Voltage level detector value selecting

Level	VldLevel1	VldLevel0	Typical voltage level
Level1	0	0	2.2
Level2	0	1	2.5
Level3	1	0	3.0
Level3	1	1	3.0

13. Interrupt Controller

The EM6617 has 12 different interrupt request sources each of which is maskable. 4 of them are coming from extarnal sources and 8 from internal.

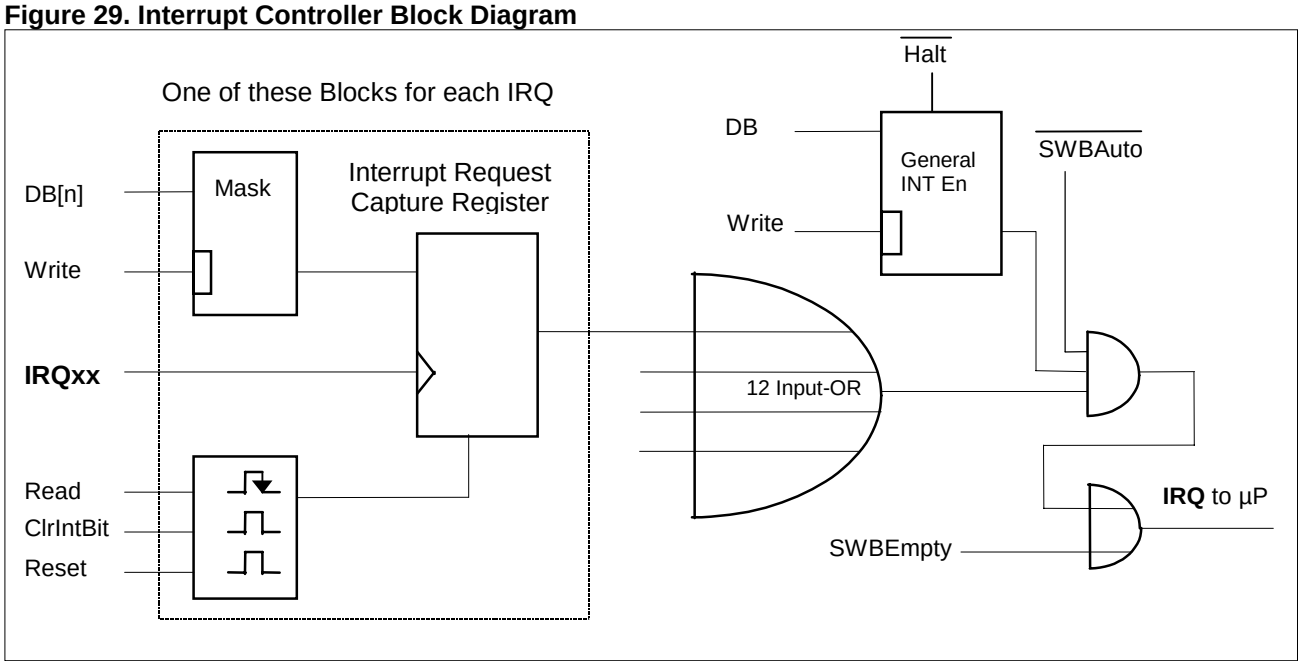
External(4)	**- Port A,**	**PA[3] .. PA[0] inputs**
Internal(8)	**- Prescaler**	**Ck[1], 32 Hz / 8 Hz**
	- 10-bit Counter	**Count to 0, Count equal to Compare**
	- EEPROM	**End of writing operation**
	- ADC	**End of conversion**
	- VLD	**End of measure**
	- SWB (non-maskable)	**SWB empty in interactive mode**

Note : the interrupt request from Serial Output Buffer (SWBEmpty) in interactive mode can not be masked in opposition to the others and goes directly to the CPU.

For interrupt requests except SWBEmpty interrupt :

To be able to send an interrupt to the CPU, at least one of the interrupt request flags must '1' (**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 high by a positive edge on the **IRQxx** data flip-flop while the corresponding mask register bit (**MaskIRQxx**) is set to 1.

Figure 29. 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 IRQ to treat. Each interrupt request flag may also be reset individually by writing 1 into it .
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 the **IRQxx** registers are cleared after reading, the CPU does not miss any interrupt request which comes during the interrupt service routine. If any occurs during this time a new interrupt will be generated as soon as the software comes out of the current interrupt subroutine.

Any interrupt request sent by a periphery cell while the corresponding mask is not set will not be stored in the interrupt request register. All interrupt requests are stored in their **IRQxx** registers depending only on their mask setting and not on the general interrupt enable status.
Whenever the EM6617 goes into HALT Mode the **IntEn** bit is automatically set to 1, thus allowing to resume from halt mode with an interrupt. This behavior is blocked if SWBAuto is set high. In this case the peripheral interrupts are disabled until the SWBAuto bit is reset low. Please refer also to the SWB chapter 9.

13.1 Interrupt control registers

Table 13.1.1 register RegIRQ1

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

W*; Writing of 1 clears the corresponding bit.

Table 13.1.2 register RegIRQ2

Bit	Name	Reset	R/W	Description
3	IRQHz1	0	R/W*	Prescaler interrupt request
2	IRQHz32/8	0	R/W*	Prescaler interrupt request
1	IRQEEP	0	R/W*	EEPROM interrupt request
0	IRQADC	0	R/W*	ADC interrupt request

W*; Writing of 1 clears the corresponding bit.

Table 13.1.3 register RegIRQ3

Bit	Name	Reset	R/W	Description
3	--			
2	IRQVLD	0	R/W*	VLD 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 13.1.4 register RegIRQMask1

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

Interrupt is not stored if the mask bit is 0.

Table 13.1.5 register RegIRQMask2

Bit	Name	Reset	R/W	Description
3	MaskIRQHz1	0	R/W	Prescaler interrupt mask
2	MaskIRQHz32/8	0	R/W	Prescaler interrupt mask
1	MaskIRQEEP	0	R/W	EEPROM interrupt mask
0	MaskIRQADC	0	R/W	ADC interrupt mask

Interrupt is not stored if the mask bit is 0.

Table 13.1.6 register RegIRQMask3

Bit	Name	Reset	R/W	Description
3	--			
2	MaskIRQVLD	0	R/W	VLD 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

14. RAM

The EM6617 has two 64x4 bit RAM's built-in.

The main RAM (RAM1) is direct addressable on addresses decimal(0 to 63). A second RAM (RAM2) is indirect addressable on addresses 64,65, 66 and 67 together with the index from RegIndexAdr.

Figure 30. Ram Architecture

64 x 4 direct addressable RAM1	
RAM1_63	4 bit R/W
RAM1_62	4 bit R/W
RAM1_61	4 bit R/W
RAM1_60	4 bit R/W
. . .	. . .
RAM1_3	4 bit R/W
RAM1_2	4 bit R/W
RAM1_1	4 bit R/W
RAM1_0	4 bit R/W

64 x 4 indexed addressable RAM2		
RAM2_3	RegIndexAdr[F]	4 bit R/W
	RegIndexAdr[E]	4 bit R/W
	...	...
	RegIndexAdr[1]	4 bit R/W
	RegIndexAdr[0]	4 bit R/W
RAM2_2	RegIndexAdr[F]	4 bit R/W
	RegIndexAdr[E]	4 bit R/W
	...	...
	RegIndexAdr[1]	4 bit R/W
	RegIndexAdr[0]	4 bit R/W
RAM2_1	RegIndexAdr[F]	4 bit R/W
	RegIndexAdr[E]	4 bit R/W
	...	...
	RegIndexAdr[1]	4 bit R/W
	RegIndexAdr[0]	4 bit R/W
RAM2_0	RegIndexAdr[F]	4 bit R/W
	RegIndexAdr[E]	4 bit R/W
	...	...
	RegIndexAdr[1]	4 bit R/W
	RegIndexAdr[0]	4 bit R/W

The RAM2 addressing is indirect using the **RegIndexAdr** value as an offset to the directly addressed base **RAM2_0**, **RAM2_1** , **RAM2_2 or RAM2_3** registers.

To write or read the RAM2 the user has first to set the offset value in the **RegIndexAdr** register. The actual access then is made on the RAM2 base addresses **RAM2_0 , RAM2_1, RAM2_2** or **RAM2_3.** Refer to Figure 30. Ram Architecture, for the address mapping.

i.e. Writing hex(5) to Ram2 add location 30: First write hex(E) to RegIndexAdr, then write hex(5) to RAM2_1

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

15. Strobe Output

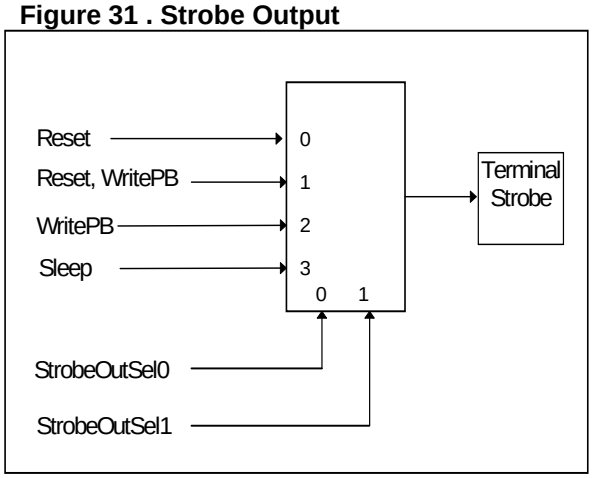
The Strobe output is used to indicate either the EM6617 reset condition, a write operation on port B (WritePB) or the sleep mode. The selection is done in register **RegLcdCntl1**. Per default, the reset condition is output on the Strobe terminal.
For a port B write operation the strobe signal goes high for half a system clock period. Data can be latched on the falling edge of the strobe signal. This function is used to indicate when data on port B output terminals is changing.
The reset signal on the Strobe output is a copy of the internal CPU reset signal. The Strobe pin remains active high as long as the CPU gets the reset.
Both the reset condition and the port B write operation can be output simultaneously on the Strobe pin.

The strobe output select latches are reset by initial power on reset only.

Table 13.1.1. Strobe Output Selection

StrobeOutSel1	StrobeOutSel0	Strobe Terminal Output
0	0	System Reset
0	1	System Reset and WritePB
1	0	WritePB
1	1	Sleep

Figure 31 . Strobe Output



15.1 Strobe register

Table 15.1.1 register OPTPCandStr

Bit	Name	Power on value	R/W	Description
3	*NoPdPC*	*0*	*R/W*	*no pull-down on PortC*
2	*NchOpDPC*	*0*	*R/W*	*N-Channel Open Drain on PortC*
1	StrobeOutSel1	0	R/W	Strobe output selection
0	StrobeOutSel0	0	R/W	Strobe output selection

Default : System reset on STROBE output

16. PERIPHERAL MEMORY MAP

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

Register Name	Add Hex	Add Dec	Reset Value	Read Bits	Write Bits	Remarks
			b'3210	Read/Write_bits		
Ram1_0	00	0	xxxx	0: Data0 1: Data1 2: Data2 3: Data3		Direct addressable Ram 64x4 bit
...	...	...	...	...		...
Ram1_63	3F	63	xxxx	0: Data0 1: Data1 2: Data2 3: Data3		Direct addressable Ram 64x4 bit
Ram2_0	40	64	xxxx	0: Data0 1: Data1 2: Data2 3: data3		16 nibbles addressable over index register on add 'H70
...	...	...	...	...		...
Ram2_3	43	67	xxxx	0: Data0 1: data1 2: Data2 3: Data3		16 nibbles addressable over index register on add 'H70
---	44	68				Reserved, not implemented
...	...	...				...
---	4F	79				Reserved, not implemented
RegPA	50	80	xxxx	0: PAData[0] 1: PAData[1] 2: PAData[2] 3: PAData[3]	----	Read port A directly
RegPBCntl	51	81	0000	0: PBIOCntl[0] 1: PBIOCntl[1] 2: PBIOCntl[2] 3: PBIOCntl[3]		Port B control Default: input mode
RegPBData	52	82	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]	Port B data output Pin port B read Default : 0
RegPCCntl	53	83	0000	0: PCIOCntl 1: '0' 2: '0' 3: '0'	0: PCIOCntl 1: -- 2: -- 3: --	Port C control Default: input mode
RegPCData	54	84	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]	Port C data output Pin port C read Default : 0
RegSWBCntl	55	85	0000	0: SWBFSel0 1: SWBFSel1 2: '0' 3: '0'	0: SWBFSel0 1: SWBFSel1 2: -- 3: --	SWB control : Clock selection
RegSWBuff	56	86	1111	0: Buff[0] 1: Buff[1] 2: Buff[2] 3: Buff[3]		SWB buffer register

Register Name	Add Hex	Add Dec	Reset Value	Read Bits	Write Bits	Remarks
			b'3210	Read/Write_bits		
RegSWBSizeL	57	87	0000	0: Size[0] 1: Size[1] 2: Size[2] 3: Size[3]		SWB size low bits
RegSWBSizeH	58	88	0000	0: Size[4] 1: Size(5] 2: SWBStart 3: SWBAuto		SWB size high bits Automatic/interactive mode
RegEEPCntl	59	89	0000	0: EEPAdr[4] 1: EEPRdWr 2: EEPBusy 3: EEPage	0: EEPAdr[4] 1: EEPRdWr 2: -- 3: EEPage	EEPROM control : Address high bit, read/write and busy flag
RegEEPAdr	5A	90	0000	0: EEPAdr[0] 1: EEPAdr[1] 2: EEPAdr[2] 3: EEPAdr[3]		EEPROM address low bits
RegEEPDataL	5B	91	0000	0: EEPData[0] 1: EEPdata[1] 2: EEPData[2] 3: EEPData[3]		EEPROM data low bits
RegEEPDataH	5C	92	0000	0: EEPData[4] 1: EEPData[5] 2: EEPData[6] 3: EEPData[7]		EEPROM data high bits
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 : Load, compare, event counter and start
RegCDataL	5F	95	1111	0: Count[0] 1: Count[1] 2: Count[2] 3: Count[3]	0: CReg[0] 1: CReg[1] 2: CReg[2] 3: CReg[3]	10 bit counter Data low nibble
RegCDataM	60	96	1111	0: Count[4] 1: Count[5] 2: Count[6] 3: Count[7]	0: CReg[4] 1: CReg[5] 2: CReg[6] 3: CReg[7]	10 bit counter Data middle nibble
RegCDataH	61	97	0011	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 nibble
RegADCCntl	62	98	0000	0: ChannelSelA 1: ChannelSelB 2: Single 3: ADCbusy	0: ChannelSelA 1: ChannelSelB 2: Single 3: StartConvert	ADC control : Channel, mode selection Start and busy flag
RegADCDataL	63	99	0000	0: ADCData[0] 1: ADCData[1] 2: ADCData[2] 3: ADCData[3]		ADC data low nibble
RegADCDataH	64	100	0000	0: ADCData[4] 1: ADCData[5] 2: ADCData[6] 3: ADCData[7]		ADC data high nibble

Register Name	Add Hex	Add Dec	Reset Value	Read Bits	Write Bits	Remarks
			b'3210	Read/Write Bits		
RegIRQMask1	65	101	0000	0: MaskIRQPA[0] 1: MaskIRQPA[1] 2: MaskIRQPA[2] 3: MaskIRQPA[3]		Port A interrupt mask Masking active low
RegIRQMask2	66	102	0000	0: MaskIRQADC 1: MaskIRQEEP 2: MaskIRQHz32/8 3: MaskIRQHz1		Prescaler, EEPROM, ADC interrupt mask Masking active low
RegIRQMask3	67	103	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	68	104	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: Port A IRQ Write: Reset IRQ if data bit = 1.
RegIRQ2	69	105	0000	0: IRQADC 1: IRQEEP 2: IRQHz32/8 3: IRQHz1	0: RIRQADC 1: RIRQEEP 2: RIRQHz32/8 3: RIRQHz1	Read: Prescaler, EEPROM, ADC IRQ ; Write: Reset IRQ if data bit = 1
RegIRQ3	6A	106	0000	0:IRQCntComp 1: IRQCount0 2: IRQVLD 3: '0'	0: RIRQCntComp 1: RIRQCount0 2: RIRQVLD 3: --	Read: 10 bit counter, VLD IRQ Write: Reset IRQ if data bit =1.
RegSysCntl1	6B	107	00x0	*0: ChTmDis* 1: SelIntFull 2: '0' 3: IntEn	*0: ChTmDis* 1: SelIntFull 2: Sleep 3: IntEn	System control 1 : *ChTmDis only usable only for EM test modes with Test=1*
RegSysCntl2	6C	108	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	6D	109	0000	0: NoLogicWD 1: NoOscWD 2: '0' 3: Vref1/2Sel	0: NoLogicWD 1: NoOscWD 2: -- 3: Vref1/2Sel	System control 3 : Watchdogs control, Reference Voltage for ADC
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
RegIndexAdr	70	112	0000	0: IndexAdr[0] 1: IndexAdr[1] 2: IndexAdr[2] 3: IndexAdr[3]		Index addressing register for4x16 nibble of Ram2
RegPresc	71	113	0000	0: DebSel 1: PrIntSel 2: '0' 3: PWMOn	0: DebSel 1: PrIntSel 2: ResPresc 3: PWMOn	Prescaler control : Debouncer and prescaler interrupt selection
RegVldCntl	72	114	0000	0: VLDlevel0 1: VLDlevel1 2: VLDBusy 3: VLDResult	0: VLDlevel0 1: VLDlevel1 2: VLDStart 3: --	VLD control : Level detection start (busy flag) and result

17. Option Register Memory Map

The values of the option registers are set by initial reset on power up and through write operations only. Other resets as reset from watchdog, reset from input port A 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		
OPTDebIntPA OPT[3:0]	75	117	0000	0: NoDebIntPA[0] 1: NoDebIntPA[1] 2: NoDebIntPA[2] 3: NoDebIntPA[3]		Option register : Debouncer on port A for interrupt gen. Default: debouncer on
OPTIntEdgPA OPT[7:4]	76	118	0000	0: IntEdgPA[0] 1: IntEdgPA[1] 2: IntEdgPA[2] 3: IntEdgPA[3]		Option register : Interrupt edge select on port A Default: pos edge
OPTNoPullPA OPT[11:8]	77	119	0000	0: NoPullPA[0] 1: NoPullPA[1] 2: NoPullPA[2] 3: NoPullPA[3]		Option register : Pull-down selection on port A Default: pull-down
OPTNoPdPB OPT[15:12]	78	120	0000	0: NoPdPB[0] 1: NoPdPB[1] 2: NoPdPB[2] 3: NoPdPB[3]		Option register : Pull-down selection on port B Default: pull-down
OPTNchOpDPB OPT[19:16]	79	121	0000	0: NchOpDPB[0] 1: NchOpDPB[1] 2: NchOpDPB[2] 3: NchOpDPB[3]		Option register : Nch. open drain output on port B Default: CMOS output
OPTPCandStr OPT[23:20]	7A	122	0000	0: StrobeOutSel0 1: StrobeOutSel1 2: NchOpDPC 3: NoPdPC		Strobe output selection Nch. open drain output Pull-down selection on port C
OPTSelPB OPT[31:28]	7B	123	0000	0: InpResSleep 1: PB2kHzOut 2: PB32kHzOut 3: PB1kHzOut		Port A input reset option Option register : Frequency output on port B
OPTInpRSel1	7C	124	0000	0: InpRes1PA[0] 1: InpRes1PA[1] 2: InpRes1PA[2] 3: InpRes1PA[3]		Option register : Reset through port A inputs selection, refer to reset part
OPTInpRSel2	7D	125	0000	0: InpRes2PA[0] 1: InpRes2PA[1] 2: InpRes2PA[2] 3: InpRes2PA[3]		Option register; Reset through port A inputs selection, refer to reset part
RegTestEM	*7F*	*127*	----	----	*Accu*	*For EM test only;* *Write accu on port B* *Test = 1*

18. 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).

19. Mask Options

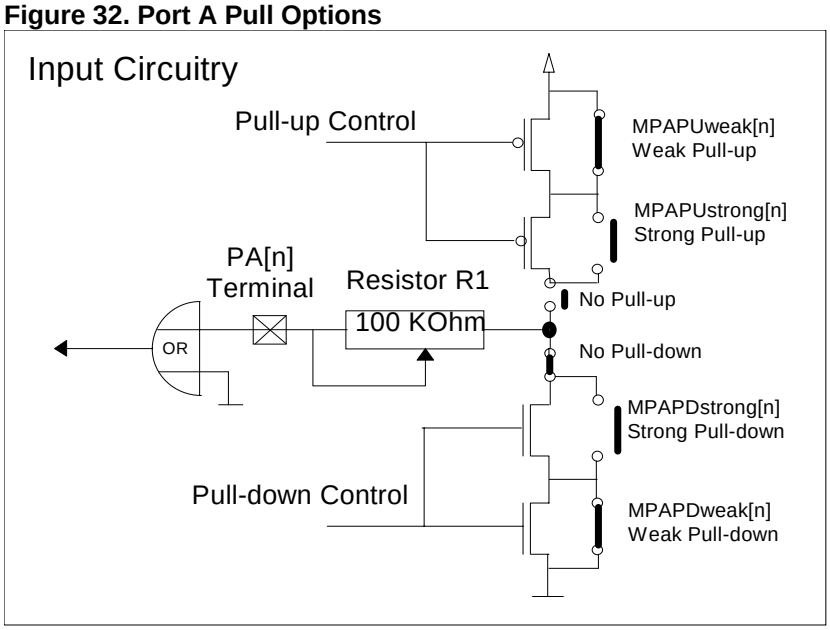
Most options which in many µControllers are realized as metal mask options are directly user selectable with the option registers, therefore allowing a maximum freedom of choice .
The following options can be selected at the time of programming the metal mask ROM.

19.1 Input / Output Ports

19.1.1 Port A Metal Options

Pull-up or no pull-up can be selected for each port A input. A pull-up selection is excluding a pull-down on the same input.
Pull-down (default) or no pull-down can be selected for each port A 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. Weak, strong or none must be chosen. The default is strong. The default resistor R1 value is 100 KOhm. The user may choose a different value from 150 KOhm down to 0 Ohm. However the value must first be checked and agreed by EM Microelectronic Marin SA.

Figure 32. Port A Pull Options



Option Name		Strong Pull-down	W Pull-down	R1 Value Typ.100 k	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 an **X** in column 1,2 and 4 and reconfirm the R1 value in column 3.

The default value is : Strong pull-down with R1=100 KOhm

Option Name		Strong Pull-up	Weak Pull-up	R1 Value typ.100k	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 an **X** in column 1,2 and 4 and reconfirm the R1 value in column 3.

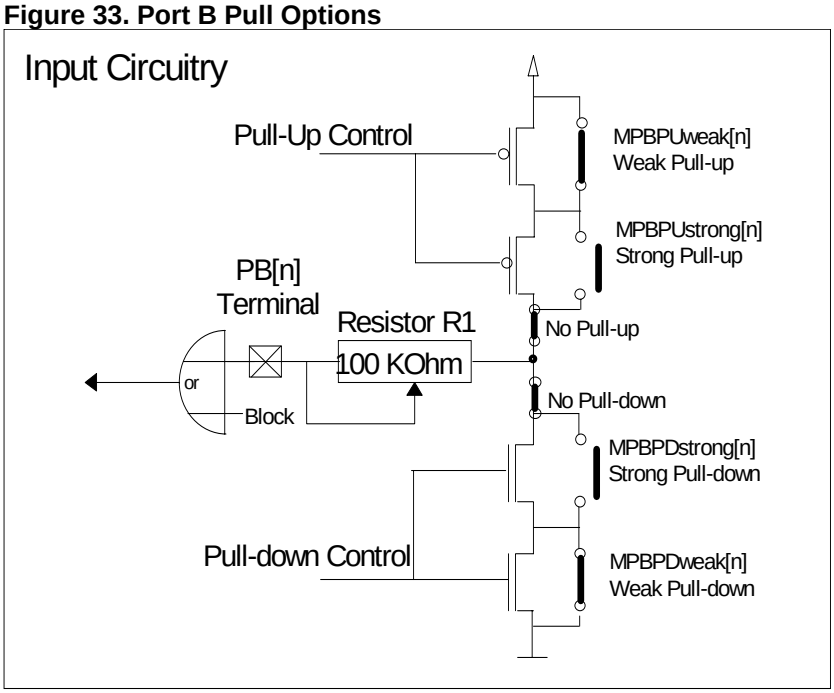
The default value is : No pull-up

19.1.2 Port B Metal Options

Pull-up or no pull-up can be selected for each port B input. The pull-up is only active in Nch. open drain mode. Pull-down or no pull-down can be selected for each port B 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. Weak , strong or none must be chosen. The default is strong. The default resistor R1 value is 100 KOhm. The user may choose a different value from 150 KOhm down to 0 Ohm. However the value must first be checked and agreed by EM Microelectronic Marin SA

Figure 33. Port B Pull Options



Option Name		Strong Pull-down	Weak Pull-down	R1 Value Typ.100k	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 an **X** in column 1,2 and 4 and reconfirm the R1 value in column 3.

The default value is : Strong pull-down with R1=100 KOhm

Option Name		Strong Pull-up	Weak Pull-up	R1 value Typ. 100k	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 an **X** in column 1,2 and 4 and reconfirm the R1 value in column 3.

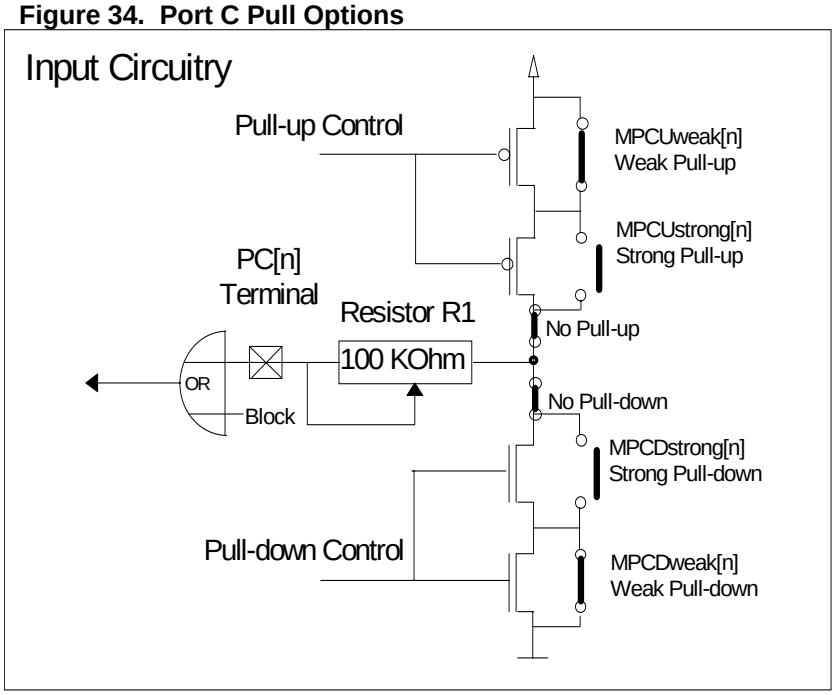
The default value is : Strong pull-up with R1=100 KOhm

19.1.3 Port C Metal Options

Pull-up or no pull-up can be selected for each port C input. The pull-up is only active in Nch. open drain mode.
Pull-down or no pull-down can be selected for each port C 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. Weak , strong or none must be chosen. The default is strong. The default resistor R1 value is 100 KOhm. The user may choose a different value from 150 KOhm down to 0 Ohm. However the value must first be checked and agreed by EM Microelectronic Marin SA. Refer also to chapter

Figure 34. Port C Pull Options



Option Name		Strong Pull-down	Weak Pull-down	R1 Value Typ.100k	NO Pull-Down
		1	**2**	**3**	**4**
MPCD[3]	PC3 input pull-down				
MPCD[2]	PC2 input pull-down				
MPCD[1]	PC1 input pull-down				
MPCD[0]	PC0 input pull-down				

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

The default value is : strong pull-down with R1=100 KOhm

Option Name		Strong Pull-up	weak Pull-up	R1 Value Typ. 100k	NO Pull-up
		1	**2**	**3**	**4**
MPCU[3]	PC3 input pull-up				
MPCU[2]	PC2 input pull-up				
MPCU[1]	PC1 input pull-up				
MPCU[0]	PC0 input pull-up				

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

The default value is : strong pull-up with R1=100 KOhm

19.1.4 SWB high impedance state

Option Name		NO	Yes
		A	**B**
MSWBZ_Clk	SWB Clock high impedance state.		

By default the SWB Clock output is driven to logic '0' outside a transmission. The user may choose high impedance state on Clock output instead of logic'0'.

Option Name		NO	Yes
		A	**B**
MSWBZ_Dat	SWB Data high impedance state.		

By default the SWB Data output is driven to logic '0' outside a transmission. The user may choose high impedance state on Data output instead of logic'0'.

19.1.5 Debouncer Frequency Option

Option Name		Ck[11]	Ck[14]
		A	**B**
MDeb	Debouncer freq.		

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

19.1.6 System Frequency

Option Name		32kHz	128kHz
		A	**B**
MFreq	System frequency = Xtal frequency		

By default the system frequency is defined as being 32kHz. Higher Frequencies are possible. A second setting guarantees typical write times for the EPROM at a system frequency of 128 kHz.

19.1.7 Additional mask options

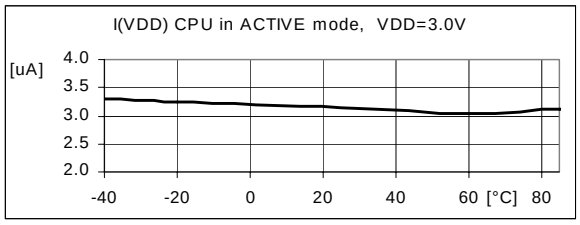
Other functions and parameters may also be changed using the metal 1 mask (i.e SVLD levels). Please contact EM Marin if you have a special request.

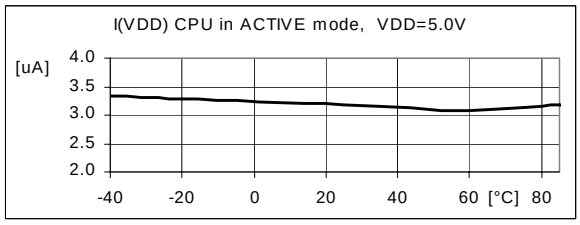
The customer should specify the required options at the time of ordering. A copy of the pages 48 to 51 as well as the « Software ROM characteristic file » generated by the assembler (*.STA) should be attached to the order. Also the Customer package marking should be defined at that time.

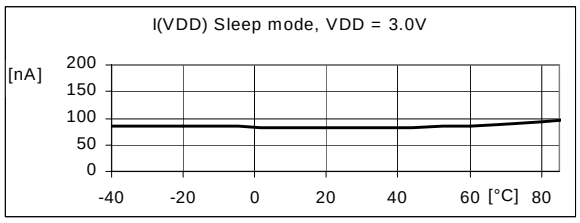
20. Temp. and Voltage Behavior

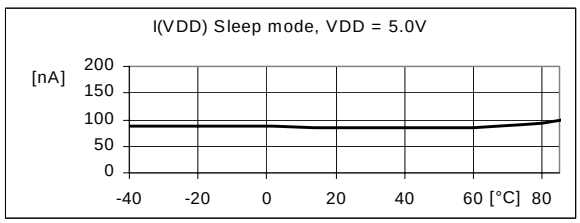
20.1 I(VDD) Current

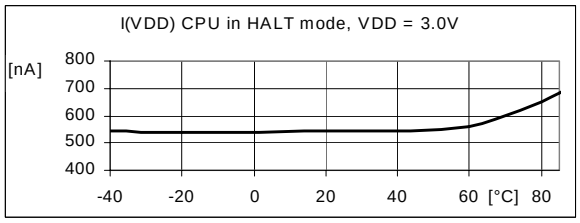
I(VDD) current over temperature for run mode, halt mode, sleep mode, ADC and EEPROM running modes.

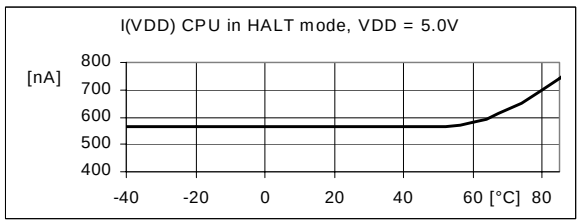


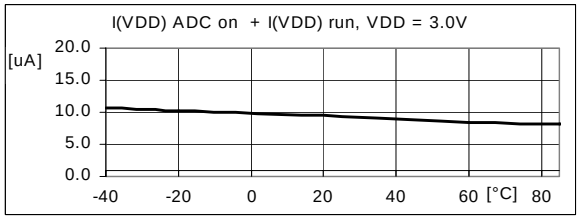


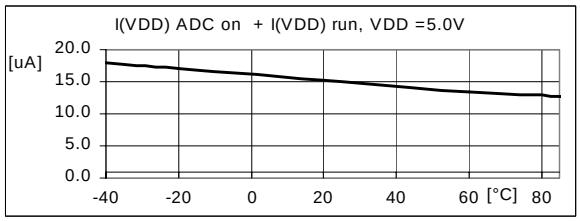


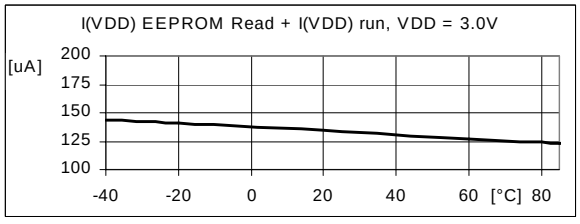


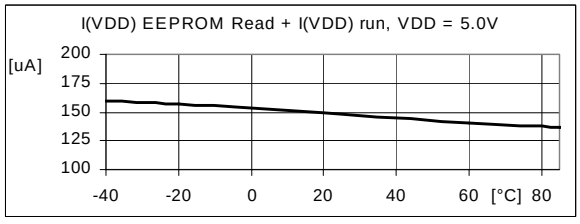


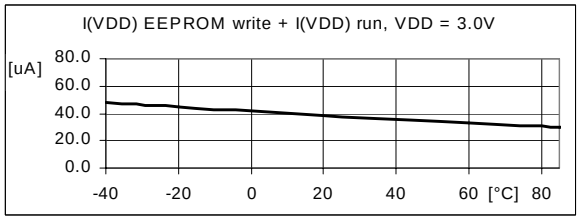


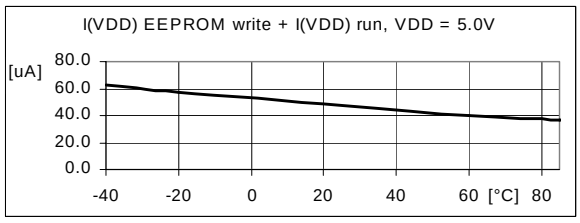






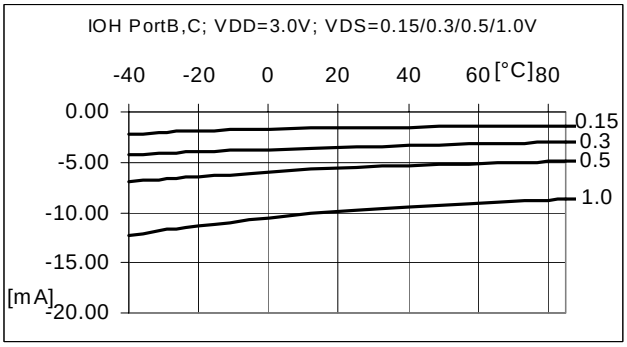


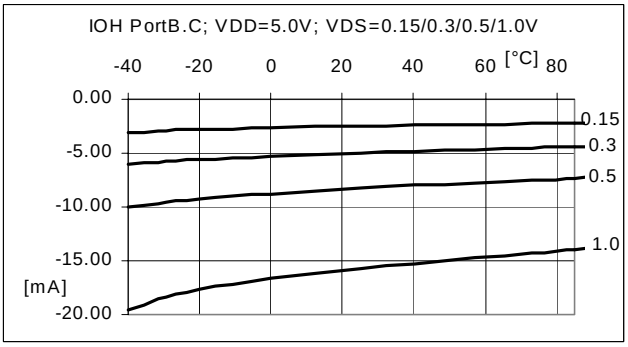


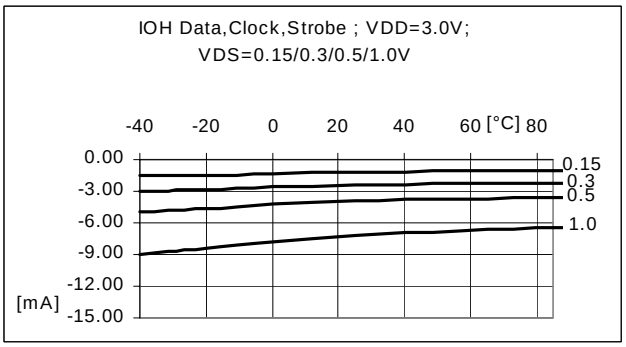


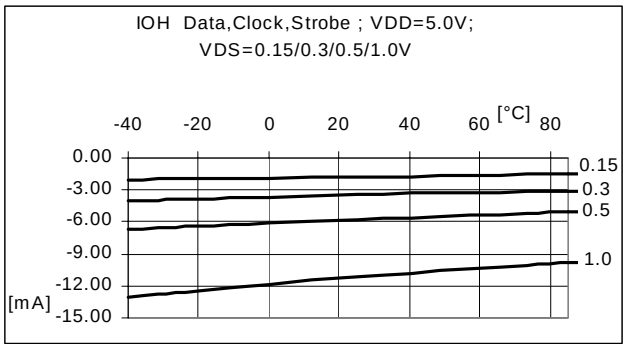
20.2 IOL, IOH

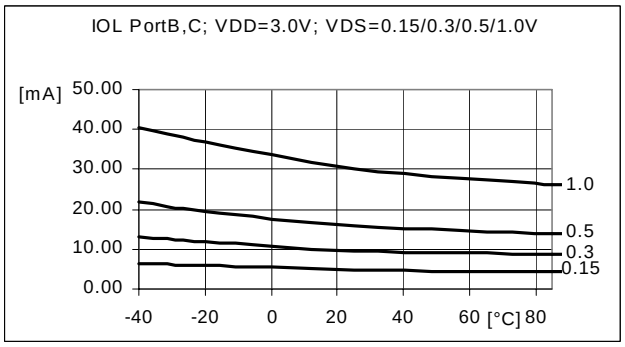
IOL and IOH temperature dependencies on different VDS voltages for port B, port C, Strobe, Clk, Data.

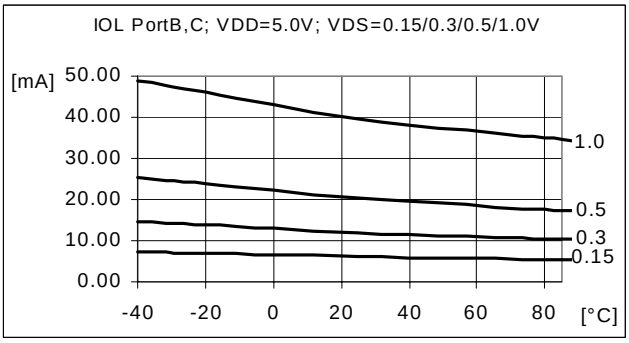


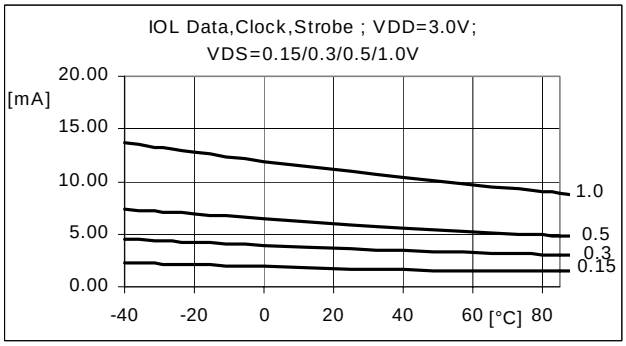


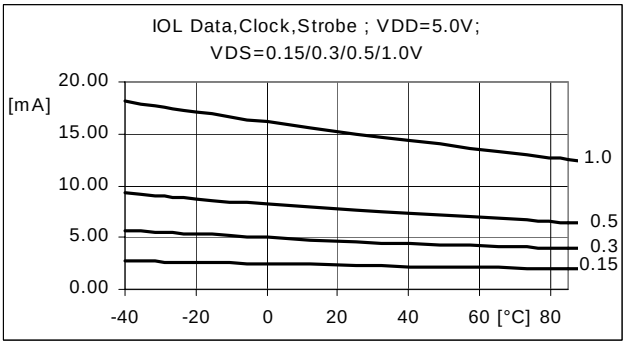






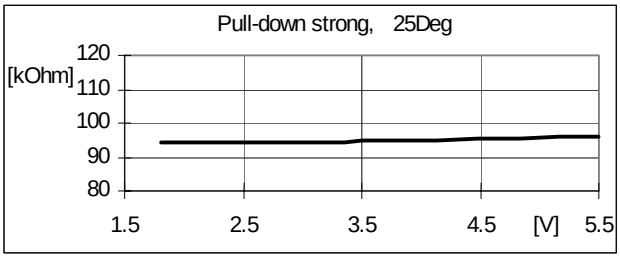


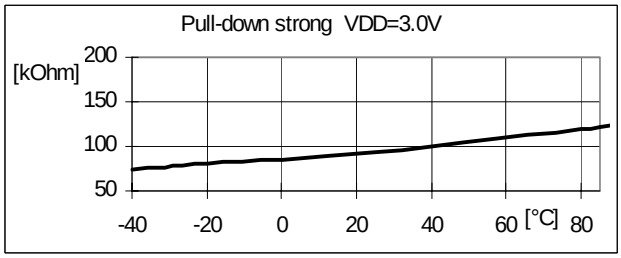


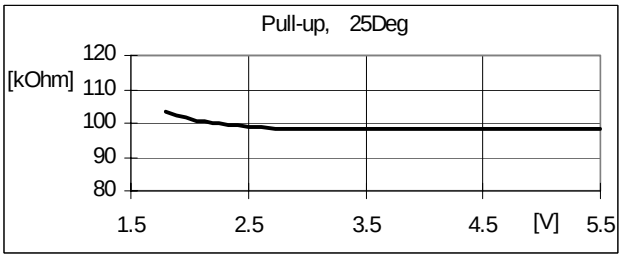


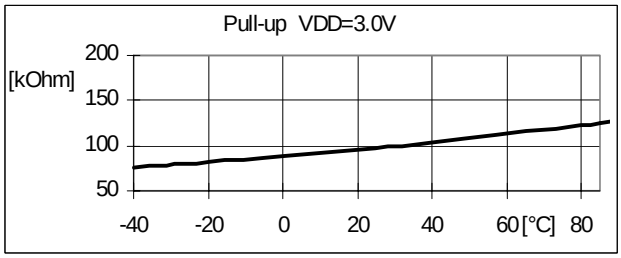
20.3 Pull-up, Pull-down

Pull-up and pull-down temperature and voltage dependencies for port A, port B, port C.



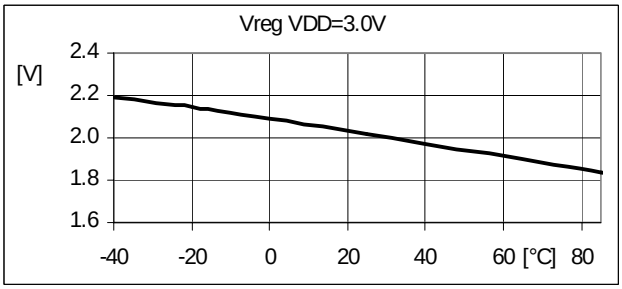


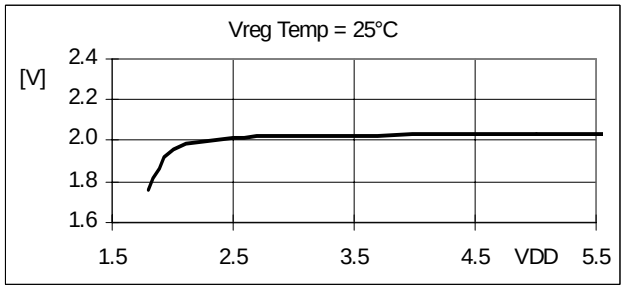


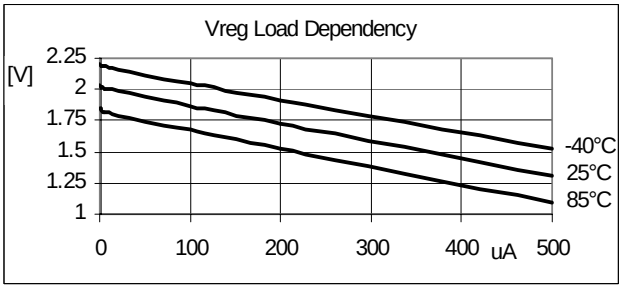


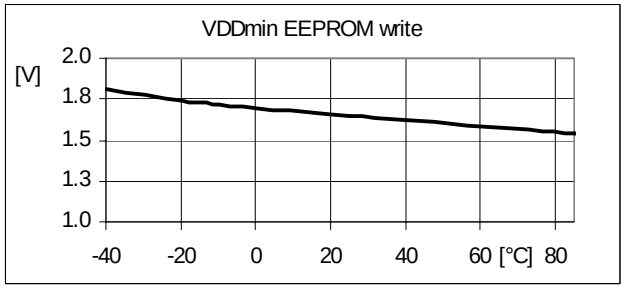
20.4 Vreg, EEPROM

Vreg voltage, temperature and load behavior. EEPROM minimal VDD characteristic over temperature.



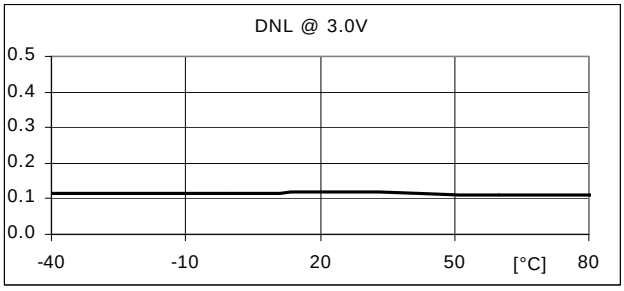


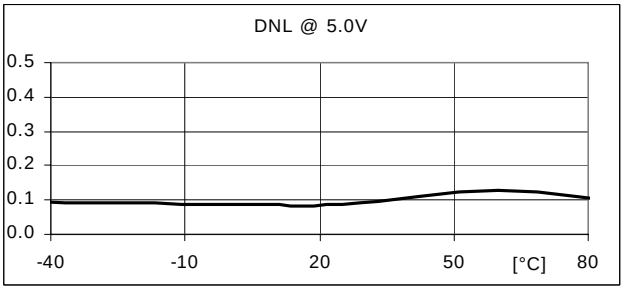


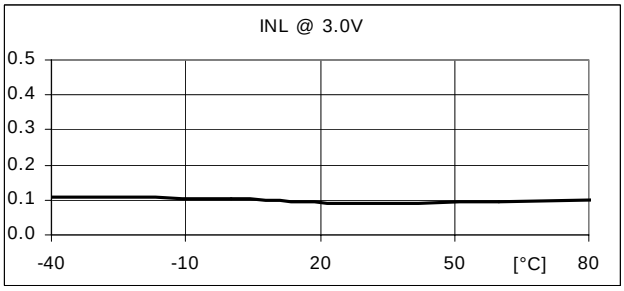


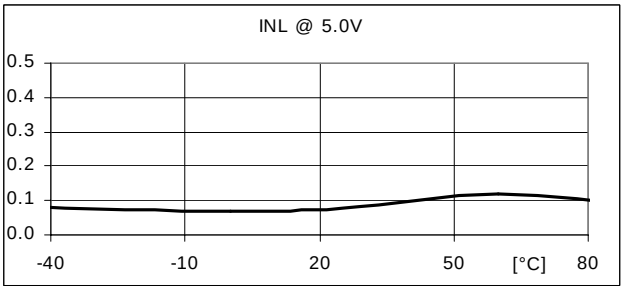
20.5 ADC8

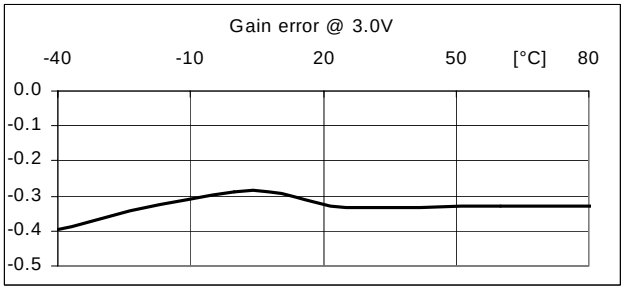
Differential, integral non-linearity, gain, offset and total unadjusted errors

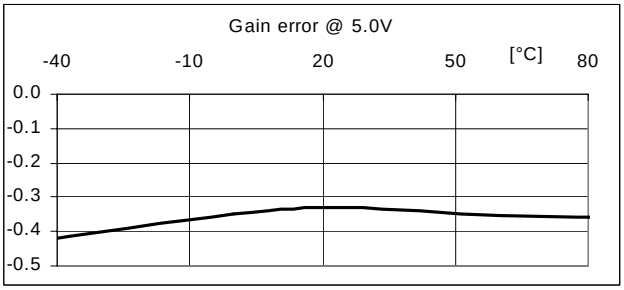


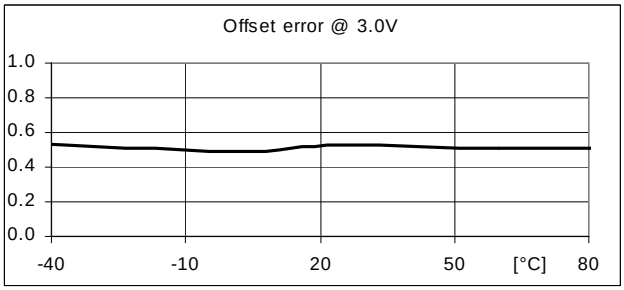


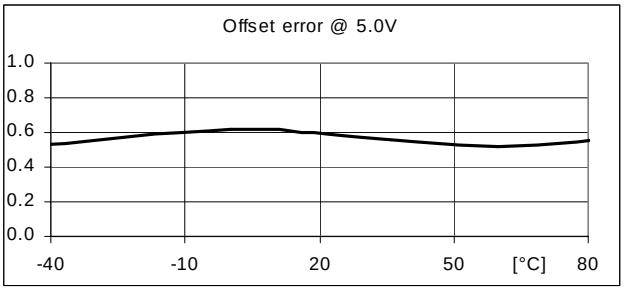


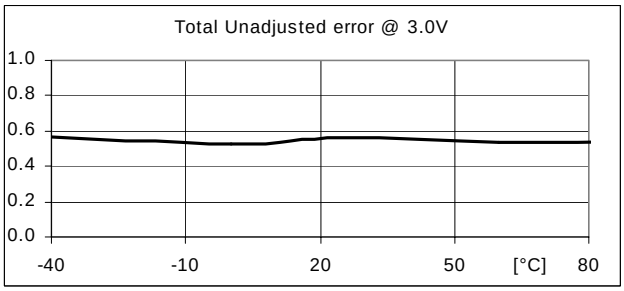


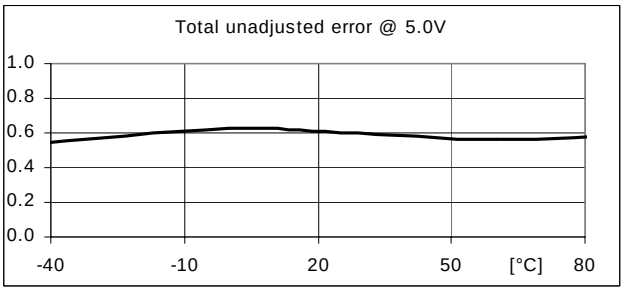


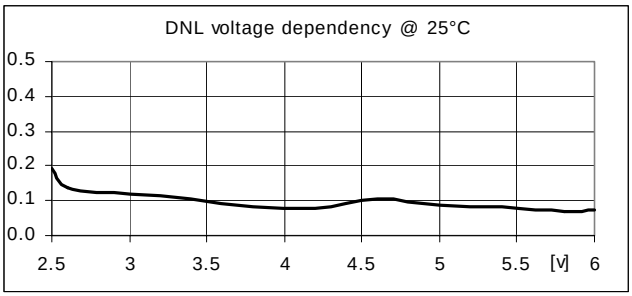


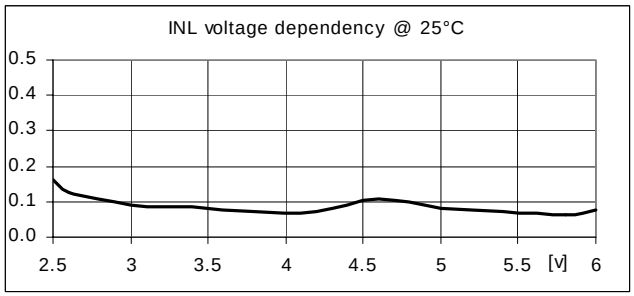


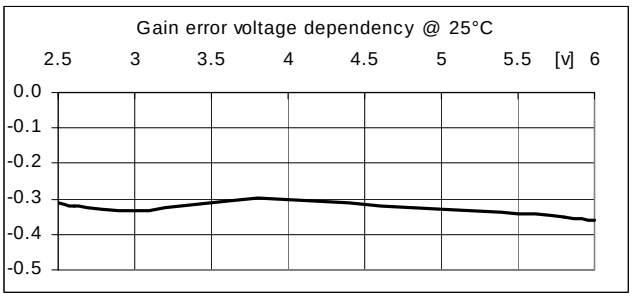


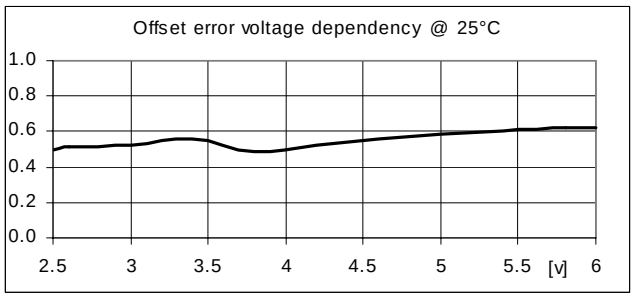


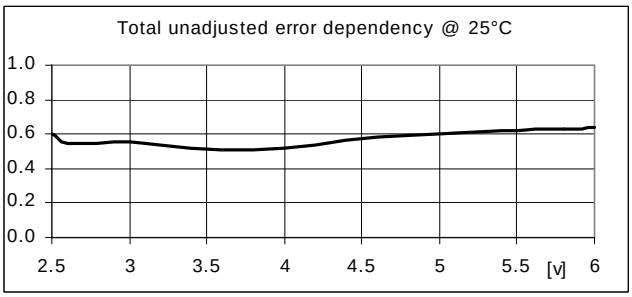


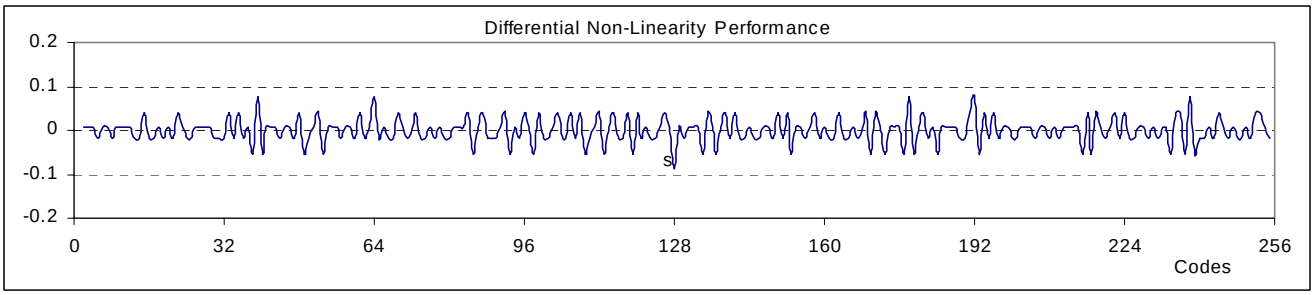


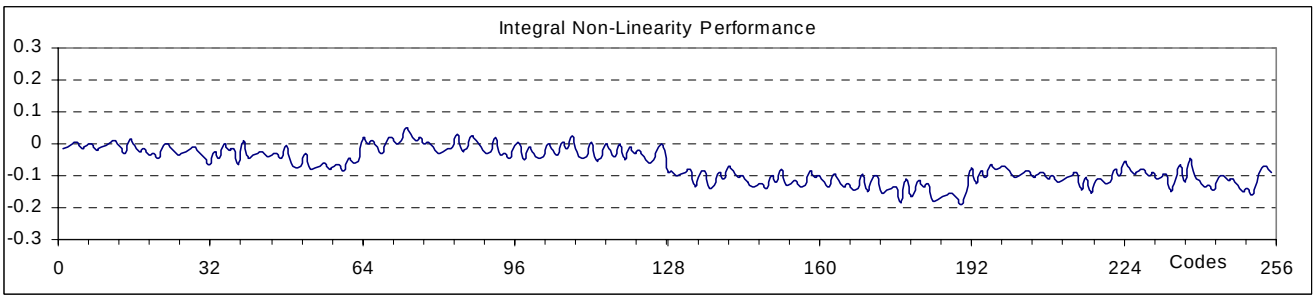












21. Electrical Specification

21.1 Absolute Maximum Ratings

	Min.	Max.	Units
Power supply V_{DD}-V_{SS}	- 0.2	+ 6.5	V
Input voltage	V_{SS} - 0,2	V_{DD}+0,2	V
Storage temperature	- 40	+ 125	°C
Electrostatic discharge to Mil-Std-883C method 3015.7 with ref. to V_{SS}	-2000	+2000	V
Maximum soldering conditions		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.

21.2 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.

21.3 Standard Operating Conditions

Parameter	MIN	TYP	MAX	Unit	Description
Temperature	-40	25	85	°C	
V_{DD}_Range 1 (note 1)	2.0	3.0	5.5	V	With internal voltage regulator
V_{SS}		0		V	Reference terminal, die substrate
C_{VDDCA} (note 2)	100			nF	Regulated voltage capacitor
Fq		32768		Hz	Nominal frequency
Rqs		35		KOhm	Typical quartz serial resistance
CL		8.2		pF	Typical quartz load capacitance
df/f		+/- 30		ppm	Quartz frequency tolerance

Note 1: The minimum V_{DD} of 2.0V allows proper system functionality for the core Logic and the EEPROM, However the ADC minimum voltage is > 2.6V for full range conversion and 0.2 LSB error.

Note 2: This capacitor filters switching noise from V_{DD} to keep it away from the internal logic cells. In noisy systems the capacitor should be chosen bigger than minimum value.

21.4 DC Characteristics - Power Supply

Conditions: V_{DD}=3.0V, T=25°C, 32kHz

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
ACTIVE Supply Current (in active mode , ADC on)		I_{VDDa1}		9.0	12.0	µA
	-40 ... 85°C	I_{VDDa1}			14.0	µA
ACTIVE Supply Current (in active mode , ADC off)		I_{VDDa2}		3.2	3.8	µA
	-40 ... 85°C	I_{VDDa2}			4.5	µA
EEPROM write current during write pulse (20ms)		I_{VDDa3}		38	48	µA
	-40 ... 85°C	I_{VDDa3}			70	µA
EEPROM read current during read pulse (30µs)		I_{VDDa4}		90	120	µA
	-40 ... 85°C	I_{VDDa4}			150	µA
STANDBY Supply Current (in Halt mode, ADC off)		I_{VDDh1}		0.55	0.8	µA
	-40 ... 85°C	I_{VDDh1}			1.0	µA
SLEEP Supply Current (in sleep mode)		I_{VDDs1}		0.1	0.3	µA
	-40 ... 85°C	I_{VDDs1}			0.4	µA
POR static level		V_{POR1}	1.4	1.6	1.8	V
RAM data retention		V_{rd1}	1.4			V
Regulated voltage	V_{DD} > 2.2 V	Vreg		2.05		

21.5 Oscillator

Conditions: T=25°C , VDD=3.0V (unless otherwise specified)

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
Temperature stability	+15 ... +35 °C	df/f x dT			0,3	ppm /°C
Voltage stability	V_{DD}=2.2 - 5.5 V	df/f x dU			5	ppm /V
Input capacitor	Ref. on V_{SS}	Cin	5,5	7	8,5	pF
Output capacitor	Ref. on V_{SS}	Cout	11,0	14	17.0	pF
Transconductance	50mVpp,V_{DD}min	Gm	2.5		15.0	µA /V
Oscillator start voltage	Tstart < 10 s	Ustart	V_{DD}min			V
Oscillator start time	V_{DD} > V_{DD}Min	tdosc		0.5	3	s
System start time (oscillator + cold-start + reset)		tdsys		1.5	4	s
Oscillation detector frequency	V_{DD} > V_{DD}min	$t_{DetFreq}$			12	kHz

21.6 DC characteristics - I/O Pins

Conditions: T= -40 ... 85°C (unless otherwise specified)

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
Input Low voltage (static)						
Ports A,B,C Test	VDD = 3.0 V	VIL	Vss		0.3*VDD	V
QIN with Regulator	VDD = 3.0 V	VIL	Vss		0.1*Vreg	V
QOUT (note 7)						
Input High voltage (static)						
Ports A,B,C Test		VIH	0.7*VDD		VDD	V
QIN with Regulator		VIH	0.9*Vreg		Vreg	V
QOUT (note 7)						
Output Low Current	VDD=3.0V , VOL=0.15V	IOL		4.8		mA
PortB. PortC	VDD=3.0V , VOL=0.30V	IOL		9.5		mA
	VDD=3.0V , VOL=0.50V	IOL		15.0		mA
	VDD=3.0V , VOL=1.0V	IOL	16.0	29.0		mA
Output Low Current	VDD=3.0V , VOL=0.15V	IOL		1.8		mA
Data, Clock, Strobe	VDD=3.0V , VOL=0.30V	IOL		3.6		mA
	VDD=3.0V , VOL=0.50V	IOL		5.8		mA
	VDD=3.0V , VOL=1.0V	IOL	6.0	10.0		mA
Output High Current	VDD=3.0V, VOH= VDD-0.15V	IOH		-1.6		mA
PortB, PortC	VDD=3.0V, VOH= VDD-0.30V	IOH		-3.2		mA
	VDD=3.0V, VOH= VDD-0.50V	IOH		-5.4		mA
	VDD=3.0V, VOH= VDD-1.0 V	IOH		-10	-6.0	mA
Output High Current	VDD=3.0V, VOH= VDD-0.15V	IOH		-1.3		mA
Data, Clock, Strobe	VDD=3.0V, VOH= VDD-0.30V	IOH		-2.5		mA
	VDD=3.0V, VOH= VDD-0.50V	IOH		-4.1		mA
	VDD=3.0V, VOH= VDD-1.0 V	IOH		-7.5	-4	mA
Input Pull-down Test, Reset	VDD=3.0V, Pin at 3.0V, 25°C	RPD		12k		Ohm
Input Pull-down Port A,B,C (note 8) weak	VDD=3.0V, Pin at 3.0V, 25°C	RPD		150k		Ohm
Input Pull-up Port A,B,C (note 8) weak	VDD=3.0V, Pin at 0.0V, 25°C	RPU		400k		Ohm
Input Pull-down Port A,B,C (note 8) strong	VDD=3.0V, Pin at 3.0V, 25°C	RPD	63k	94k	142k	Ohm
Input Pull-up Port A,B,C (note 8) strong	VDD=3.0V, Pin at 0.0V, 25°C	RPU	62k	98k	146k	Ohm

Note 7 ; QOUT (OSC2) is used only with Quartz.
Note 8 : Weak or strong are standing for weak pull or strong pull transistor. Values are for R1=100kOhm

21.7 Supply Voltage Level Detector

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
SVLD voltage Level1	-10 ... 60°C	V_{SVLD1}	2.02	2.20	2.38	V
	-40 ... 85°C		1.98	2.20	2.42	
SVLD voltage Level2	-10 ... 60°C	V_{SVLD2}	2.30	2.50	2.70	V
	-40 ... 85°C		2.25	2.50	2.75	
SVLD voltage Level3	-10 ... 60°C	V_{SVLD3}	2.78	3.02	3.26	V
	-40 ... 85°C		2.72	3.02	3.32	
Temperature coefficient	0 ... 50°C			< +/- 0.2		mV/°C

21.8 ADC 8 Bit

Conditions: T=25°C , VDD=3.0V, System Clock=32 KHz , Ramp Input

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
Differential nonlinearity	Vgnd = VDD/2 Vref = VDD	DNL		+/-0.2	+/-0.5	LSB
Integral nonlinearity	Vgnd = VDD/2 Vref = VDD	INL		+/-0.2	+/-0.5	LSB
Gain Error	Vgnd = VDD/2 Vref = VDD	GE		+/- 0.5		LSB
Offest Error	Vgnd = VDD/2 Vref = VDD	OE		+/- 0.5		LSB
Total unadjusted error	Vgnd = VDD/2 Vref = VDD	TUE		+/- 1		LSB
Battery voltages for full range conversion	Error DNL 0.2 LSB	Vrange	2.6		5.5	V
Missing Codes			No Missing Codes			

The total unadjusted error is a combination of the offset, gain and INL errors.

21.9 EEPROM

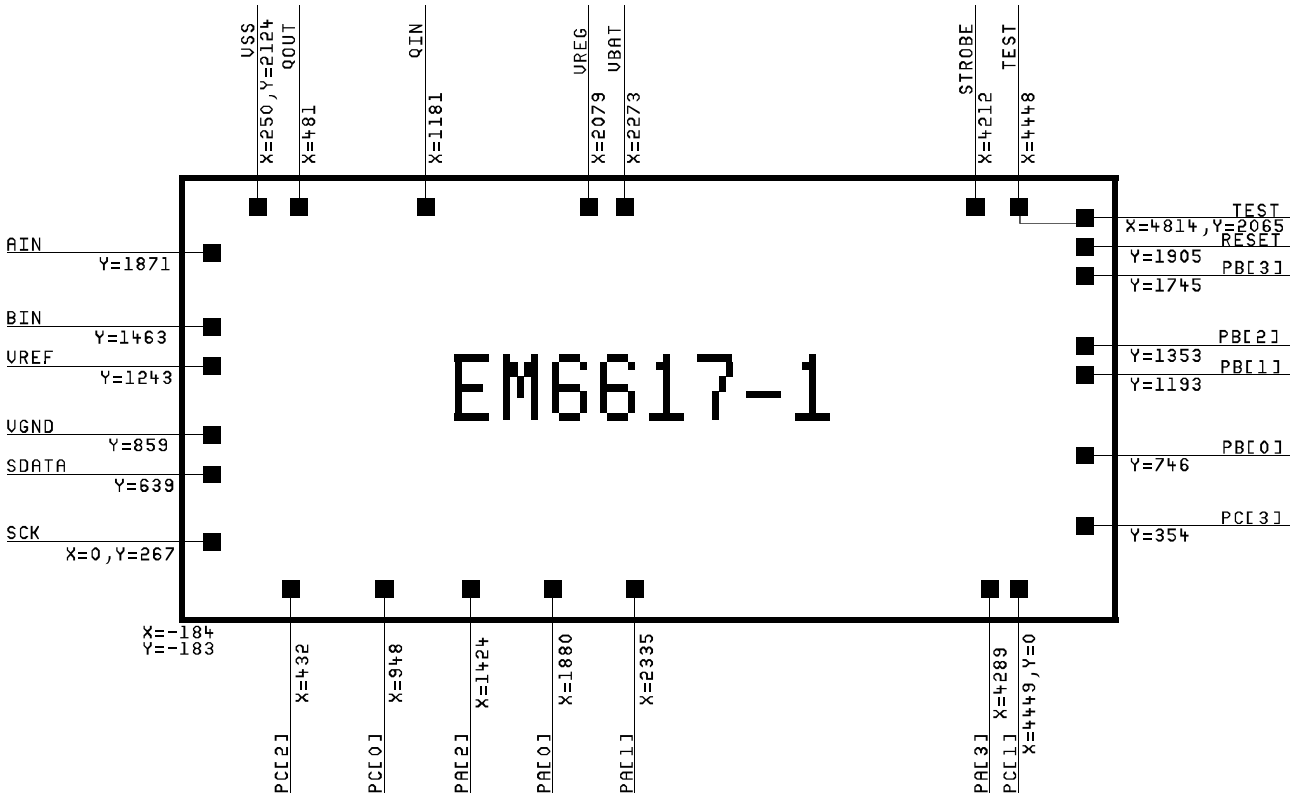
Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
Read time (note 9)	-40 ... 85°C	EEPrd		45		us
Write time (note 9)	-40 ... 85°C	EEPwr		24		ms
VDD during write and read operation	-40 ... 85°C	V_{EEP}	2.0		5.5	V

Note 9 : The typical values are guaranteed by design for a) 32kHz operation or b) 128kHz operation with the corresponding metal mask set. Using different frequencies one must assure not to fall below min value.

22. Pad Location Diagram

The Pad Test is twice on the circuit, it can be bound either on top or on the left side. Internally the two pads test are connected together by a metal wire.

Figure 35. Die Pad Location

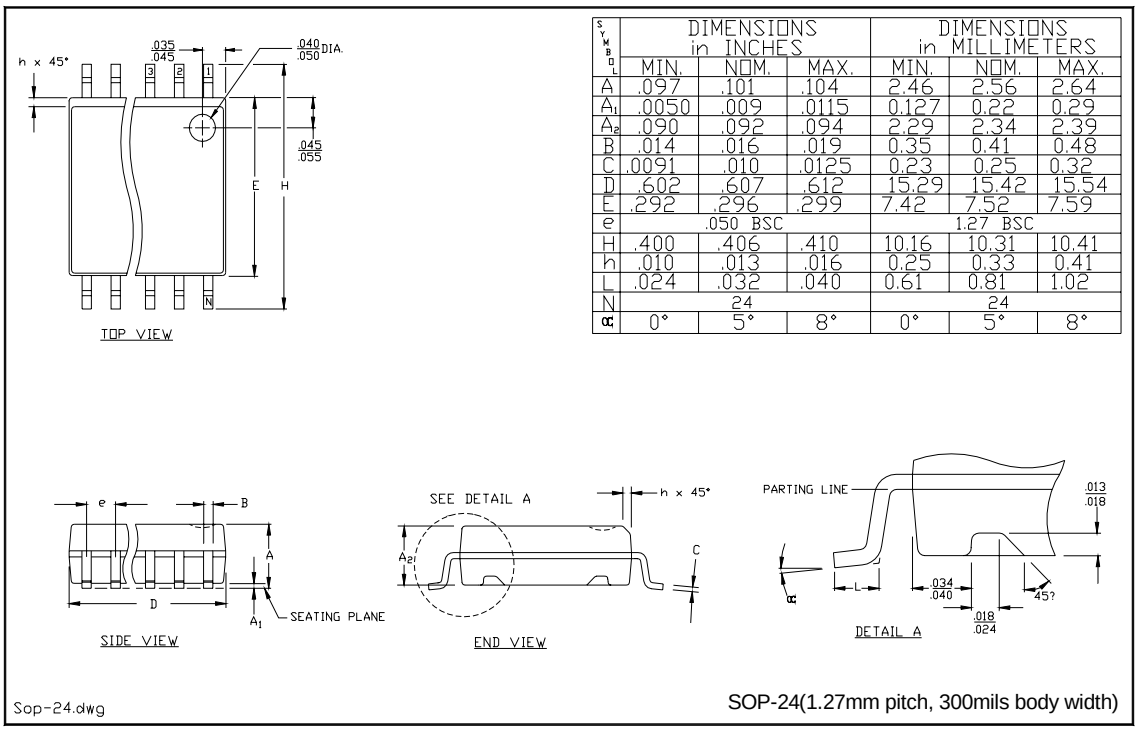


CHIP SIZE is X = 5181 by Y = 2489 Microns or X = 204 by Y = 98 Mils

NOTE : THE ORIGIN (0,0) IS THE LOWER LEFT COORDINATE OF CENTER PADS
THE LOWER LEFT CORNER OF THE CHIP SHOWS DISTANCES TO ORIGIN

23. Package & Ordering information

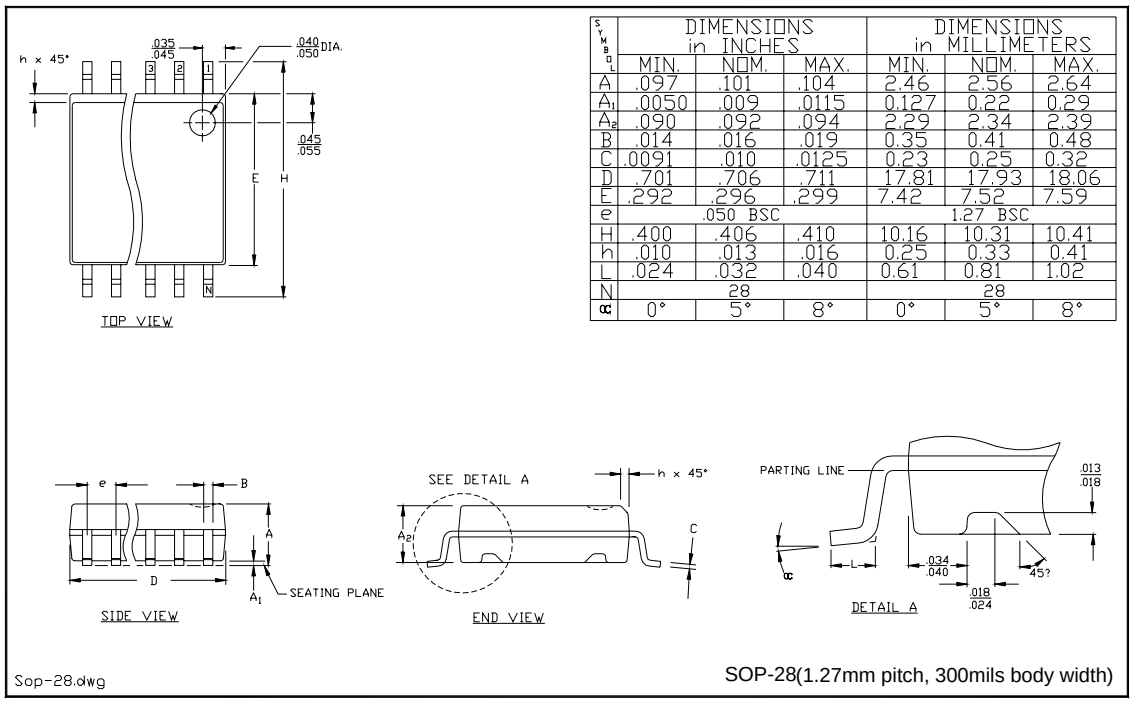
Figure 36. SOP 24 Pin Package



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	.602	.607	.612	15.29	15.42	15.54
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		24			24	
α	0°	5°	8°	0°	5°	8°

SOP-24(1.27mm pitch, 300mils body width)

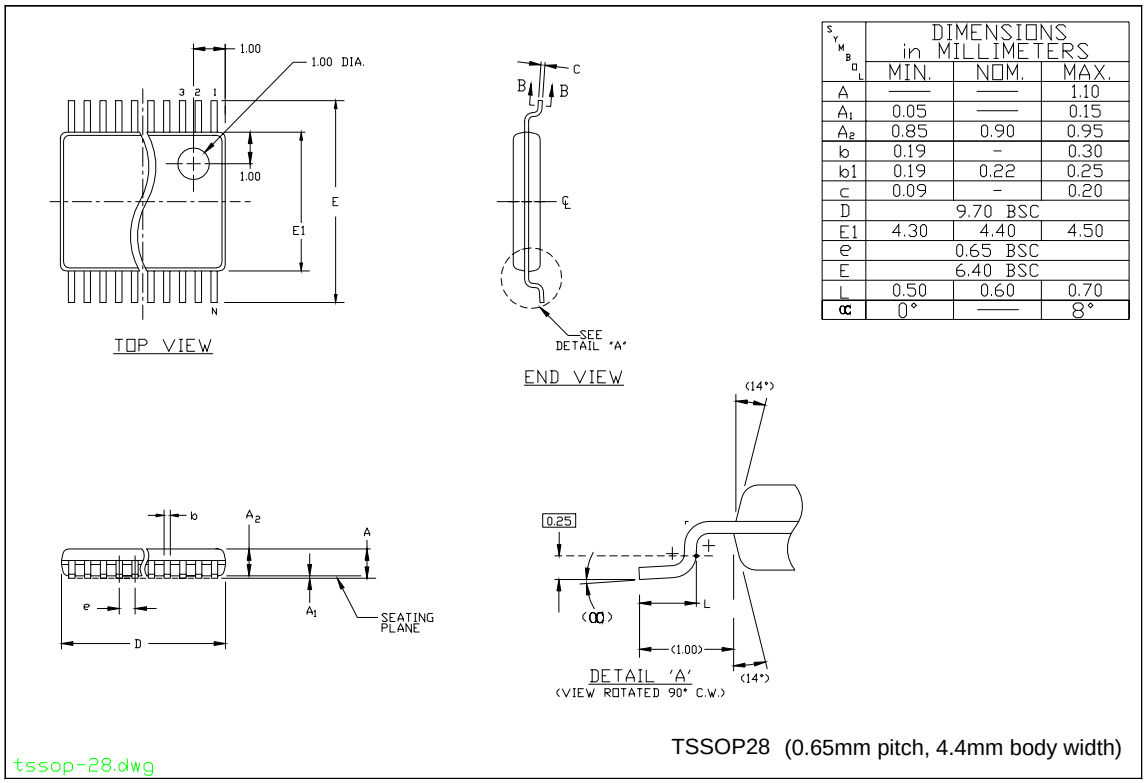
Figure 37. SOP 28 Pin Package



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	.701	.706	.711	17.81	17.93	18.06
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		28			28	
α	0°	5°	8°	0°	5°	8°

SOP-28(1.27mm pitch, 300mils body width)

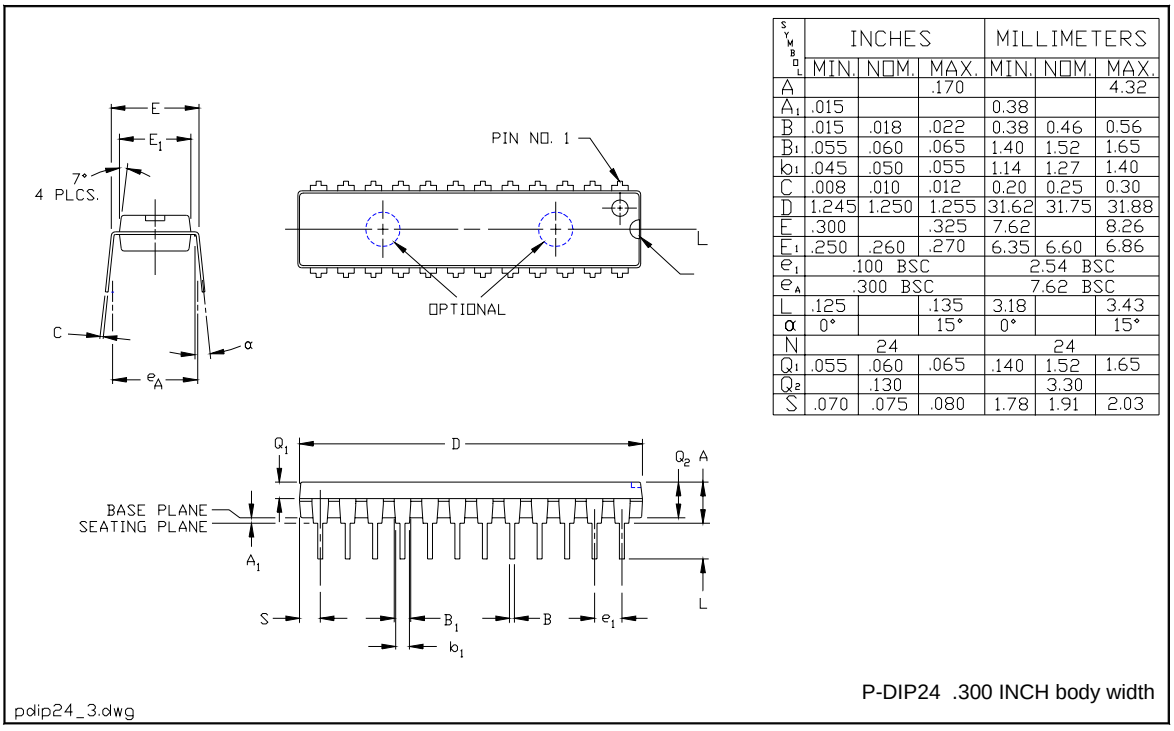
Figure 38. TSSOP 28 Pin Package



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	9.70 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°

TSSOP28 (0.65mm pitch, 4.4mm body width)

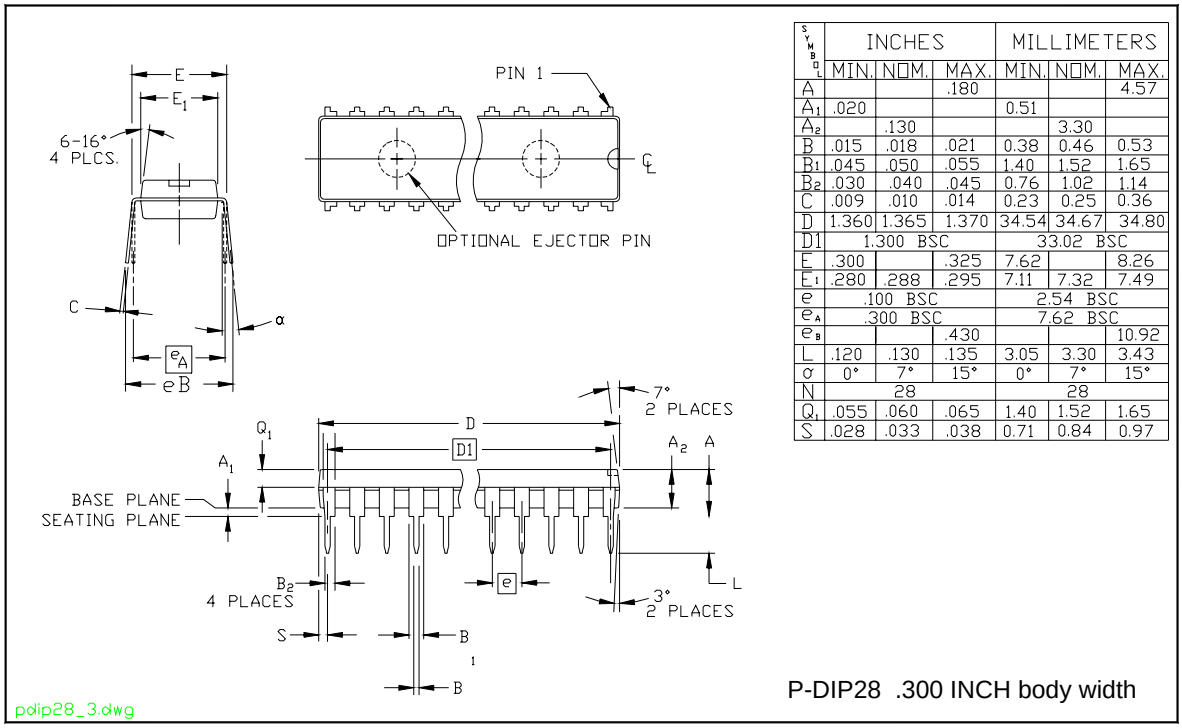
Figure 39. DIP 24 Pin Package



SYMBOL	INCHES			MILLIMETERS		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A			.170			4.32
A1	.015			0.38		
B	.015	.018	.022	0.38	0.46	0.56
B1	.055	.060	.065	1.40	1.52	1.65
b1	.045	.050	.055	1.14	1.27	1.40
C	.008	.010	.012	0.20	0.25	0.30
D	1.245	1.250	1.255	31.62	31.75	31.88
E	.300		.325	7.62		8.26
E1	.250	.260	.270	6.35	6.60	6.86
e1	.100 BSC			2.54 BSC		
eA	.300 BSC			7.62 BSC		
L	.125		.135	3.18		3.43
α	0°		15°	0°		15°
N	24			24		
Q1	.055	.060	.065	.140	1.52	1.65
Q2		.130			3.30	
S	.070	.075	.080	1.78	1.91	2.03

P-DIP24 .300 INCH body width

Figure 40. DIP 28 Pin Package



SYMBOL	INCHES			MILLIMETERS		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A			.180			4.57
A1	.020			0.51		
A2		.130			3.30	
B	.015	.018	.021	0.38	0.46	0.53
B1	.045	.050	.055	1.40	1.52	1.65
B2	.030	.040	.045	0.76	1.02	1.14
C	.009	.010	.014	0.23	0.25	0.36
D	1.360	1.365	1.370	34.54	34.67	34.80
D1	1.300 BSC			33.02 BSC		
E	.300		.325	7.62		8.26
E1	.280	.288	.295	7.11	7.32	7.49
e	.100 BSC			2.54 BSC		
eA	.300 BSC			7.62 BSC		
eB			.430			10.92
L	.120	.130	.135	3.05	3.30	3.43
α	0°	7°	15°	0°	7°	15°
N	28			28		
Q1	.055	.060	.065	1.40	1.52	1.65
S	.028	.033	.038	0.71	0.84	0.97

P-DIP28 .300 INCH body width

23.1 Ordering Information

Packaged Device:

EM6617 %%% SO28 B

Customer Version:
customer-specific number
given by EM Microelectronic

Package:
SO24, SO28 = 24/28 pin SOIC
TP28 = 28 pin TSSOP
DL24, DL28 = 24/28 pin DIP (note 1)

Delivery Form:
A = Stick
B = Tape&Reel (for SO and TP packages only)

Device in DIE Form:

EM6617 %%% 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)

Note 1: Please contact EM Microelectronic-Marin S.A. for availability of DIP package.
In the 24 pin version the port C outputs PC[3] and PC[2] are not bonded and therefore not available for the user.

Ordering Part Number (selected examples)

Part Number	Package/Die Form	Delivery Form/Thickness
EM6617%%%SO28A	28 pin SOIC	Stick
EM6617%%%SO28B	28 pin SOIC	Tape&Reel
EM6617%%%SO24A	24 pin SOIC	Stick
EM6617%%%SO24B	24 pin SOIC	Tape&Reel
EM6617%%%TP28A	28 pin TSSOP	Stick
EM6617%%%WS11	Sawn wafer	11 mils
EM6617%%%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. P01 , P12, etc.

23.2 Package Marking

DIP and SOIC marking:

First line:	E	M	6	6	1	7		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:

E	M	6	6	1	7	%	%
P	P	P	P	P	P	P	P
C	C	C	C	C	C	Y	P

Where: %% = last two-digits of the customer-specific number given by EM (e.g. 05, 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

23.3 Customer Marking

There are **11** digits available for customer marking on **PDIP24/28** and **SO24/28**.
There are **6** digits available for customer marking on **TSSOP28**.

Please specify below the desired customer marking.
