Ultra Low Power Microcontroller with 4x20 LCD Driver

Features

- **True Low Power** 1.7 µA active mode, LCD On
  0.75 µA standby mode, LCD Off
  0.1 µA sleep mode
  @ 3 V, 32 KHz, 25 ºC
- **Low Supply Voltage 1.2 V to 3.6 V**
- **Melody, 7 tones + silence inclusive 4-bit timer**
- **Universal 10-bit counter, PWM, event counter**
- LCD 20 segments, static drive, 3 or 4 times multiplexed
- Temperature compensated LCD voltage levels
- Built-in LCD voltage multipliers
- **RC Oscillator 512kHz**
- 72 basic instructions
- 2 clocks per instruction cycle
- **EM6627 Mask programmable Version 4kx16 bits**
- RAM 128 x 4 bits
- **EEPROM 8x8 bit**
- Max. 12 inputs ; port A, port B, port SP
- Max. 8 outputs ; port B, port SP
- Voltage Level Detector (VLD), 2 levels
- Prescaler down to 1 second
- 3 wire serial port , 8 bit, master and slave mode
- 5 external interrupts (port A, serial interface)
- 8 internal interrupts (3x prescaler, 2x10-bit counter, melody timer, serial interface, EEPROM)
- timer watchdog

Description

The EM6627 is an ultra-low power, low voltage microcontroller with an integrated static drive or 3/4 MUX x 20 segments LCD driver and the equivalent of 8kB program memory. It features temperature compensated LCD voltage levels, and built-in LCD voltage multiplier. It also has a melody generator, an 8x8 EEPROM and PWM function. Tools include windows-based simulator. The EM6522 programmable part can be used for must functions during program development

Due to its very low current consumption, the EM6627 is ideal for use in battery-operated and field-powered applications.

Typical Applications

- Household appliance
- Timer / sports timing devices
- Medical devices
- Interactive system with display
- Measurement equipment
- Bicycle computers
- Safety and security devices

Figure 1. Architecture

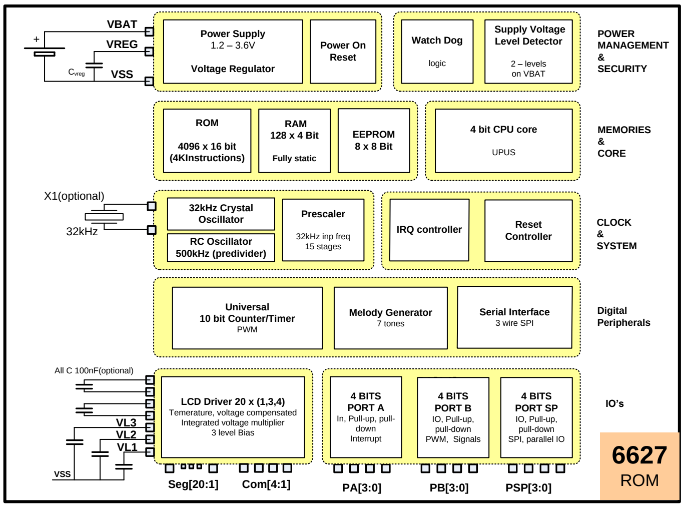
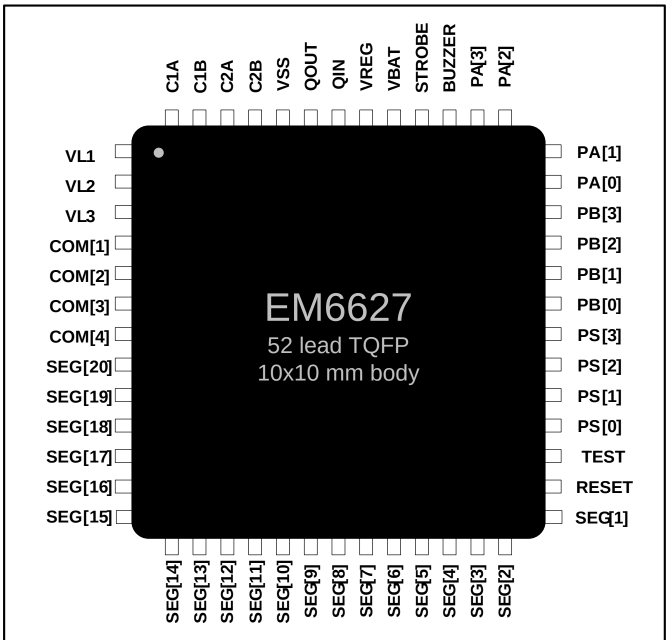


Figure 2. Pin Configuration, TQFP52



TQFP – 52 10x10 mm body

EM6627 at a glance

Power Supply
- Low voltage low power architecture including internal voltage regulator
- 1.2 V to 3.6 V battery voltage
- 1.7 uA in active mode (32kHz, LCD on, 25 °C)
- 0.75 uA in standby mode (32kHz, LCD off, 25 °C)
- 0.1 uA in sleep mode (25 °C)

CPU Clock
- Internal RC Oscillator at 512 kHz
- RC Oscillator divider by 2/4/8/16
- watch type Crystal oscillator

RAM
- 64 x 4 bit, direct addressable
- 64 x 4 bit, indexed addressable

ROM
- 4k x 16 bit (8k Byte), metal mask programmable

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 reset
- Reset terminal
- Reset with input combination on port A (register selectable)

Prescaler
- 15 stage system clock divider down to 1Hz
- 3 Interrupt requests; 1Hz, 32Hz or 8Hz, Blink
- Prescaler reset (4 kHz to 1Hz)

Liquid Crystal Display Driver (LCD)
- 20 Segments static drive, 3 or 4 times multiplexed
- Internal or external voltage multiplier
- LCD switch off for power save

8-Bit Serial Interface
- 3 wire (Clock, DataIn, DataOut) master/slave mode
- READY output during data transfer
- Maximum shift clock is equal to the main system clock
- Interrupt request to the CPU after 8 bits data transfer
- Supports different serial formats
- Can be configured as a parallel 4 bit input/output port
- Direct input read on the port terminals
- All outputs can be put tristate (default)
- Selectable pull-downs in input mode
- CMOS or Nch. open drain outputs
- Pull-up selectable in Nch. open drain mode

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)

4-Bit Bidirectionnel 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
- Pull-up in Nch. open drain mode
- Selectable PWM, 32kHz, 1kHz and 1Hz output

Melody Generator
- Dedicated Buzzer terminal
- 7 tones plus silence output
- The output can be put tristate (default)
- Internal 4-bit timer, usable also in standalone mode
- 4 different timer input clocks
- Timer with automatic reload or single run
- Timer interrupt request when reaching 0

Voltage Level Detector (SVLD)
- 2 levels
- Busy flag during measure

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

Interrupt Controller
- 5 external and 8 internal interrupt request sources
- Each interrupt request can be individually masked
- Each interrupt flag can be individually 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.

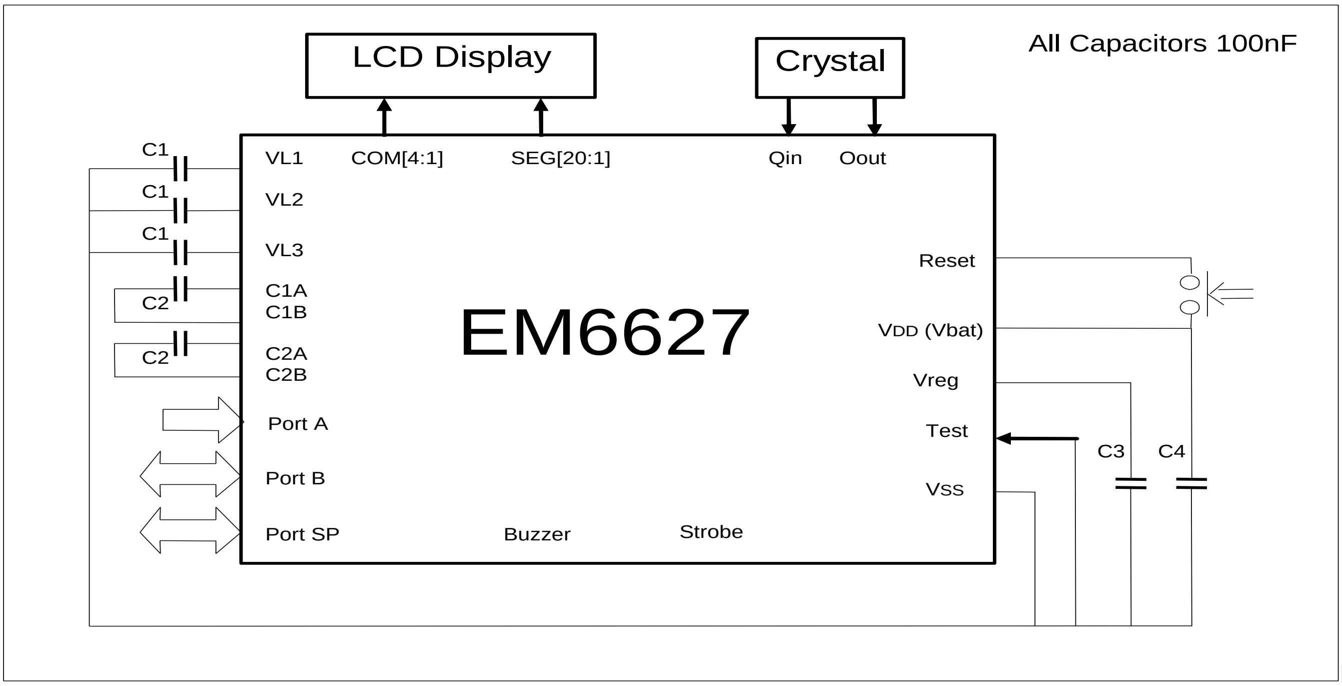
E2PROM
- 8 x 8 bit, indirect addressable
- Interrupt request at the end of a write operation
- Busy flag during read / write operation

Table of Contents

1. Pin Description for EM6627

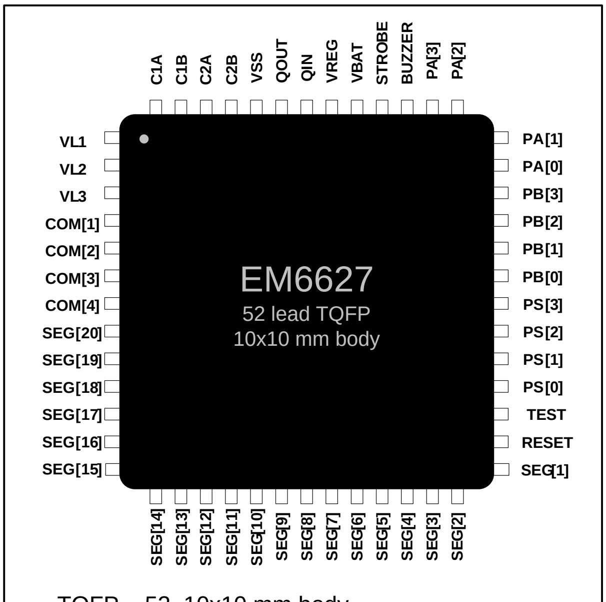
Chip	TQFP 52	DIL 64	Signal Name	Function	Remarks
13	1	10	VL1	Voltage multiplier level 1	LCD level 1 input, if ext. supply
12	2	11	VL2	Voltage multiplier level 2	LCD level 2 input, if ext. supply
11	3	12	VL3	Voltage multiplier level 3	LCD level 3 input, if ext. supply
10	4	13	COM[1]	LCD back plane 1	Only for LCD
9	5	14	COM[2]	LCD back plane 2	Only for LCD
8	6	15	COM[3]	LCD back plane 3	Only for LCD
7	7	16	COM[4]	LCD back plane 4	Only for LCD,
6	8	18	SEG[20]	LCD Segment 20	LCD or General Purpose Output
5	9	19	SEG[19]	LCD Segment 19	LCD or General Purpose Output
4	10	20	SEG[18]	LCD Segment 18	LCD or General Purpose Output
3	11	21	SEG[17]	LCD Segment 17	LCD or General Purpose Output
2	12	22	SEG[16]	LCD Segment 16	LCD or General Purpose Output
1	13	23	SEG[15]	LCD Segment 15	LCD or General Purpose Output
52	14	26	SEG[14]	LCD Segment 14	LCD or General Purpose Output
51	15	27	SEG[13]	LCD Segment 13	LCD or General Purpose Output
50	16	28	SEG[12]	LCD Segment 12	LCD or General Purpose Output
49	17	29	SEG[11]	LCD Segment 11	LCD or General Purpose Output
48	18	30	SEG[10]	LCD Segment 10	LCD or General Purpose Output
47	19	31	SEG[9]	LCD Segment 9	LCD or General Purpose Output
46	20	33	SEG[8]	LCD Segment 8	LCD or General Purpose Output
45	21	34	SEG[7]	LCD Segment 7	LCD or General Purpose Output
44	22	35	SEG[6]	LCD Segment 6	LCD or General Purpose Output
43	23	36	SEG[5]	LCD Segment 5	LCD or General Purpose Output
42	24	37	SEG[4]	LCD Segment 4	LCD or General Purpose Output
41	25	38	SEG[3]	LCD Segment 3	LCD or General Purpose Output
40	26	39	SEG[2]	LCD Segment 2	LCD or General Purpose Output
39	27	42	SEG[1]	LCD Segment 1	LCD or General Purpose Output
38	28	43	Reset	Input reset terminal,	Main reset
37	29	44	Test	Input test terminal,	For EM tests only, ground 0 !
36	30	45	PSP[0]	Input/output , open drain	Serial interface data in parallel data[0] in/out
35	31	46	PSP[1]	Output , open drain	Serial interface Ready CS parallel data[1] in/out
34	32	47	PSP[2]	Output , open drain	Serial interface data out parallel data[2] in/out
33	33	49	PSP[3]	Input/output , open drain	Serial interface clock I/O parallel data[3] in/out
32	34	50	PB[0]	Input/output, open drain	Port B data[0] I/O
31	35	51	PB[1]	Input/output, open drain	Port B data[1] I/O
30	36	52	PB[2]	Input/output, open drain	Port B data[2] I/O
29	37	53	PB[3]	Input/output, open drain	Port B data[3] I/O
28	38	54	PA[0]	Input port A terminal 0	TestVar 1, Event counter
27	39	55	PA[1]	Input port A terminal 1	TestVar 2
26	40	58	PA[2]	Input port A terminal 2	TestVar 3
25	41	59	PA[3]	Input port A terminal 3	Event counter,
24	42	60	Buzzer	Output Buzzer terminal	
23	43	61	Strobe	Output Strobe terminal	µP reset state and status output
22	44	62	Vbat = VDD	Positive power supply	
21	45	63	Vreg	Internal voltage regulator	Connect to minimum 100nF,
20	46	64	Qin/Osc1	Crystal terminal 1	32 kHz crystal,
19	47	2	Qout /Osc2	Crystal terminal 2	32 kHz crystal,
18	48	3	VSS	Negative power supply	ref. terminal,
17	49	4	C2B	Voltage multiplier	Not needed if ext. supply
16	50	5	C2A	Voltage multiplier	Not needed if ext. supply
15	51	6	C1B	Voltage multiplier	Not needed if ext. supply
14	52	7	C1A	Voltage multiplier	Not needed if ext. supply

Figure 3. Typical Configuration



o connections to QIN, Qout on versions with the internal RC Oscillator N

Figure 4. Package pinout TQFP52



TQFP – 52 10x10 mm body

Figure not to scale

2. Operating Modes

The EM6627 has two low power dissipation modes, standby and sleep. Figure 5 is a transition diagram for these modes.

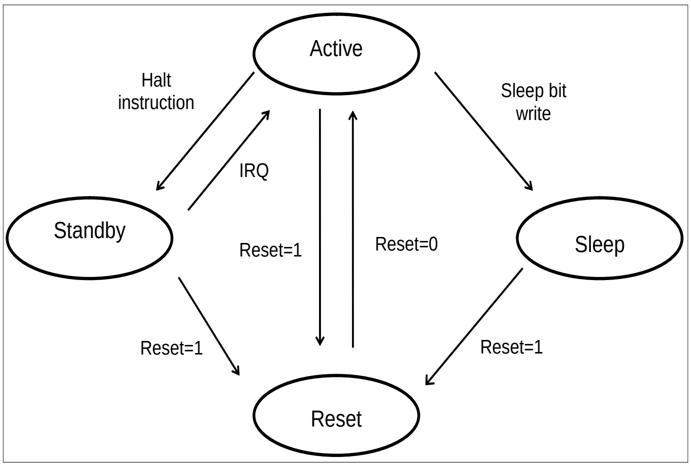
2.1 Active Mode

The active mode is the actual CPU running mode. Instructions are read from the internal ROM and executed by the CPU. 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.

2.2 Standby Mode

Executing a halt instruction puts the EM6627 into standby mode. The voltage regulator, oscillator, watchdog timer, LCD, interrupts, timers and counters are operating. However, the CPU stops since the clock related to instruction execution stops. Registers, RAM and I/O pins retain their states prior to standby mode. A reset or an interrupt request if enabled cancels standby.

Figure 5 Mode transition diagram



2.3 Sleep Mode

Writing to the **Sleep** bit in the **RegSysCntl1** register puts the EM6627 in sleep mode. The oscillator stops and most functions of the EM6627 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 EM6627 Reset terminal or by the active, selected port A input reset combination

During sleep mode and the following start up the EM6627 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 EM6627 was powered up (**SleepEn** = "0") or woken up from sleep (**SleepEn** = "1").

Before going to Sleep mode the user must assure that any pending EEPROM write is finished.

Table 2.3.1. Internal State in Standby and Sleep Mode

Function	Standby	Sleep
Oscillator	Active	Stopped
EEPROM	Active	Retains data
Instruction Execution	Stopped	Stopped
Interrupt Functions	Active	Stopped
Registers and Flags	Retained	Reset
RAM Data	Retained	Retained
Hardware Configuration 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 config registers
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

3. Power Supply

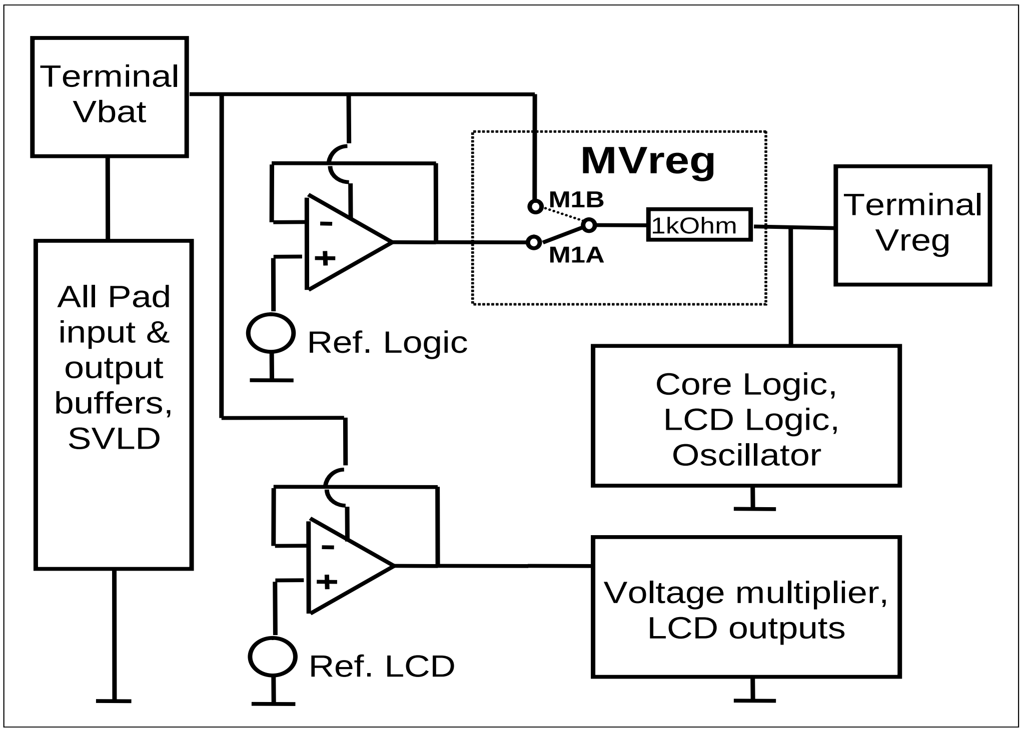
The EM6627 is supplied by a single external power supply between VDD (Vbat) and VSS (Ground). A built-in voltage regulator generates Vreg providing regulated voltage for the oscillator and the internal logic. The output drivers are supplied directly from the external supply VDD. The internal power configuration is shown below in Figure 6. Internal Power Supply.

To supply the internal core logic it is possible to use either the internal voltage regulator (Vreg < VDD) or Vbat directly (Vreg = VDD). The selection is done by metal 1 mask option. By default, the voltage regulator is used. Refer to chapter 19.1.1 for the metal mask selection.

The internal voltage regulator is chosen for high voltage systems. It saves power by reducing the internal core logic's power supply to an optimum value. However, due to the inherent voltage drop over the regulator the minimal VDD is restricted to 1.4 V .

A direct Vbat connection can be selected for systems running on a 1.5 V battery. The internal 1 KOhm resistor together with the external capacitor on Vreg is filtering the VDD supply to the internal core. In this case the minimum VDD can be as low as 1.2 V.

Figure 6. Internal Power Supply



4. Reset

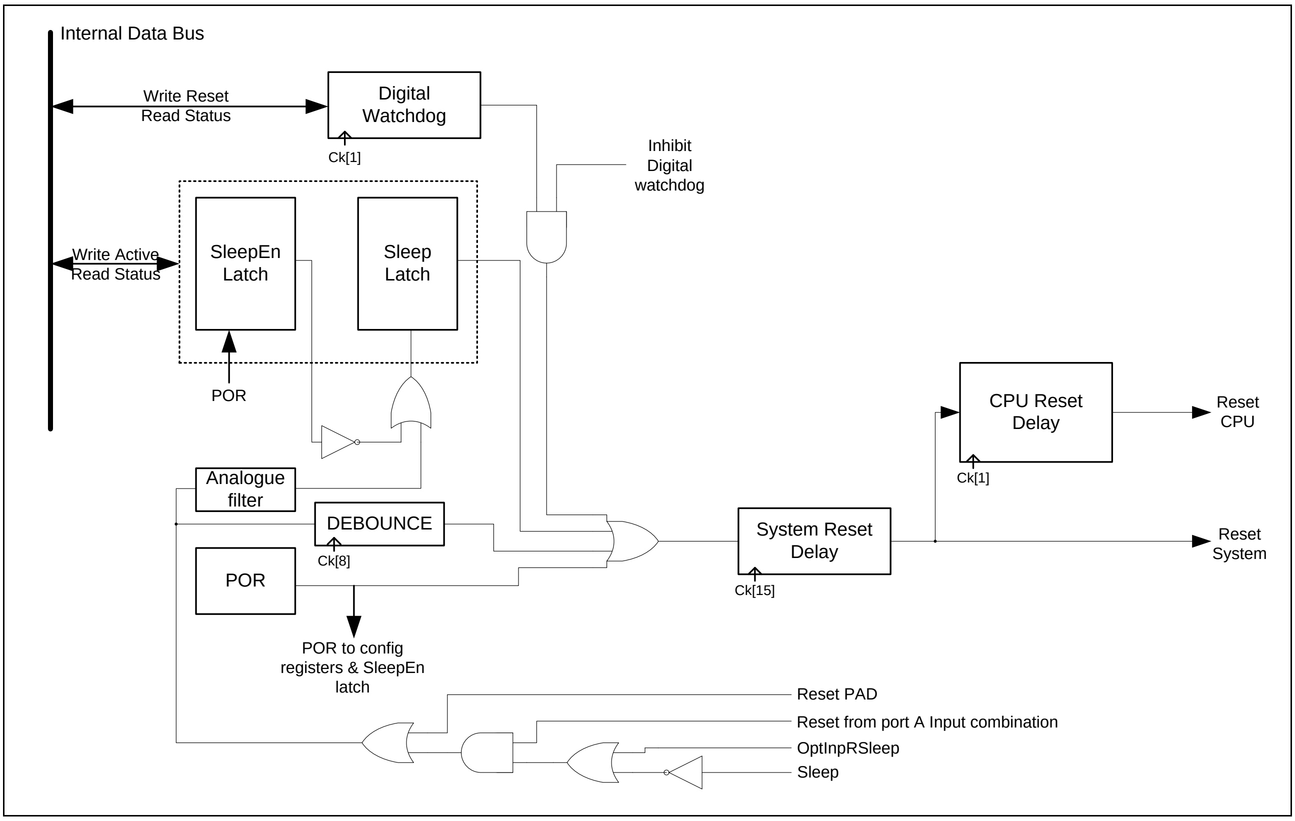
Figure 7. illustrates the reset structure of the EM6627. One can see that there are six possible reset sources :

(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 **CRegInpRSel1** and **CRegInpRSel2)**
(4) Internal reset from the Digital Watchdog. --> System Reset, Reset CPU
(5) 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 hardware configuration registers and the sleep enable (**SleepEn)** latch. System reset and reset CPU do not reset the hardware configuration registers nor the **SleepEn** latch.
CPU Reset state can be shown on Strobe terminal by selecting **StrobeOutSel1,0 = 0** in **RegLcdCntl1**.

Figure 7. Reset Structure



4.1 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.

4.2 Input Port A Reset Function

By writing the **CRegInpRSel1** and **CregInpRSel2** 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. This reset combination input is valid in active and standby mode, for Sleep mode see below.

InpResSleep bit in register CRegFSelPB 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 **CRegInpRSel1** and **CRegInpRSel2.**
Following formula is applicable :

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

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

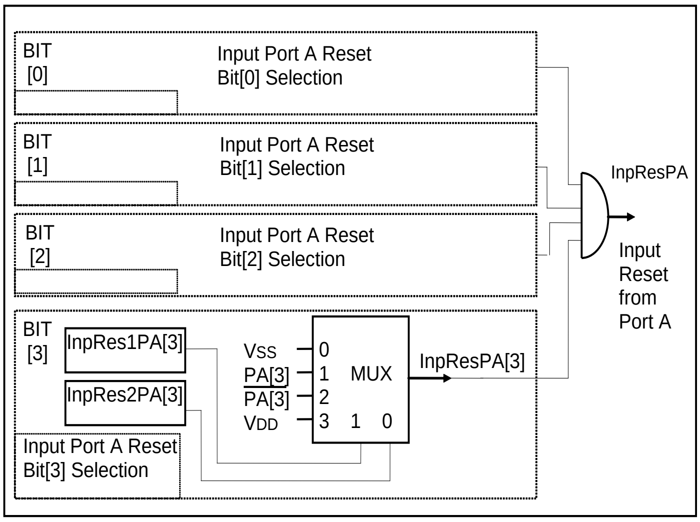
n = 0 to 3

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 8. Input Port A Reset Structure (AND-Type)



4.3 Digital Watchdog Timer Reset

The digital watchdog is a simple, non-programmable, 2-bit timer, that counts on each rising edge of Ck[1]. It will generate a system reset if it is not periodically cleared. The watchdog timer function can be inhibited by activating an inhibit digital watchdog bit (**NoLogicWD**) located in **RegVldCntl**. 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 4.3.1 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

4.4 CPU State after Reset

Reset initializes the CPU as shown in Table 4.4.1 below.

Table 4.4.1 Initial CPU Value after Reset.

Name	Bits	Symbol	Initial Value
Program counter 0	12	PC0	hex 000
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

5. Oscillator and Prescaler

5.1 Oscillator

The system clock (prescaler input) is always RC 32 kHz (derived from RC 512kHz/16 or RC 128kHz/4)
In case of RC oscillator the CPU clock is the divided RC clock. Clock selection is defined by register RCSel

Table 5.1.1 Register RegRCSel

Bit	Name	POR	R/W	Description
3	RCSel[3]	0	R/W	
2	RCSel[2]	0	R/W	
1	RCSel[1]	0	R/W	
0	RCSel[0]	0	R/W	

In case of RCFreq metal option setting MRCFreq=512kHz

RCSel[3]	RCSel[2]	RCSel[1]	RCSel[0]	CPU clock
0	0	0	0	RC 32kHz
0	0	0	1	RC 64kHz
0	0	1	x	RC128kHz
0	1	x	x	RC256kHz
1	x	x	x	RC 512kHz

The system clock is always RC based 32kHz while running on RC512kHz

In case of RCFreq metal option setting MRCFreq=128kHz

RCSel[3]	RCSel[2]	RCSel[1]	RCSel[0]	CPU clock
0	0	0	0	RC 32kHz
0	0	0	1	RC 64kHz
0	0	1	x	RC128kHz
0	1	x	x	RC128kHz
1	x	x	x	RC 128kHz

The system clock is always RC based 32kHz while running of RC128kHz

5.1.1 RC 512kHz Oscillator

If the 512kHz RC oscillator metal option is set, the user may choose to run the CPU on 512kHz, 256kHz, 128kHz, 64kHzand 32kHz. The divider selection is performed with register RegRCSel. The system frequency for all peripherals is independent of the RegRCSel selection and willl be based on the RC created 32kHz frequency.
The RC oscillator has an initial cold start delay of 0.25ms. The CPU remains in reset state during this cold start time.
The RC freq is factory pretrimmed, the trim value is located in the EEPROM and needs to be copied into the RC trim register by the CPU to take effect.

Figure 9. RC temperature and voltage dependencies

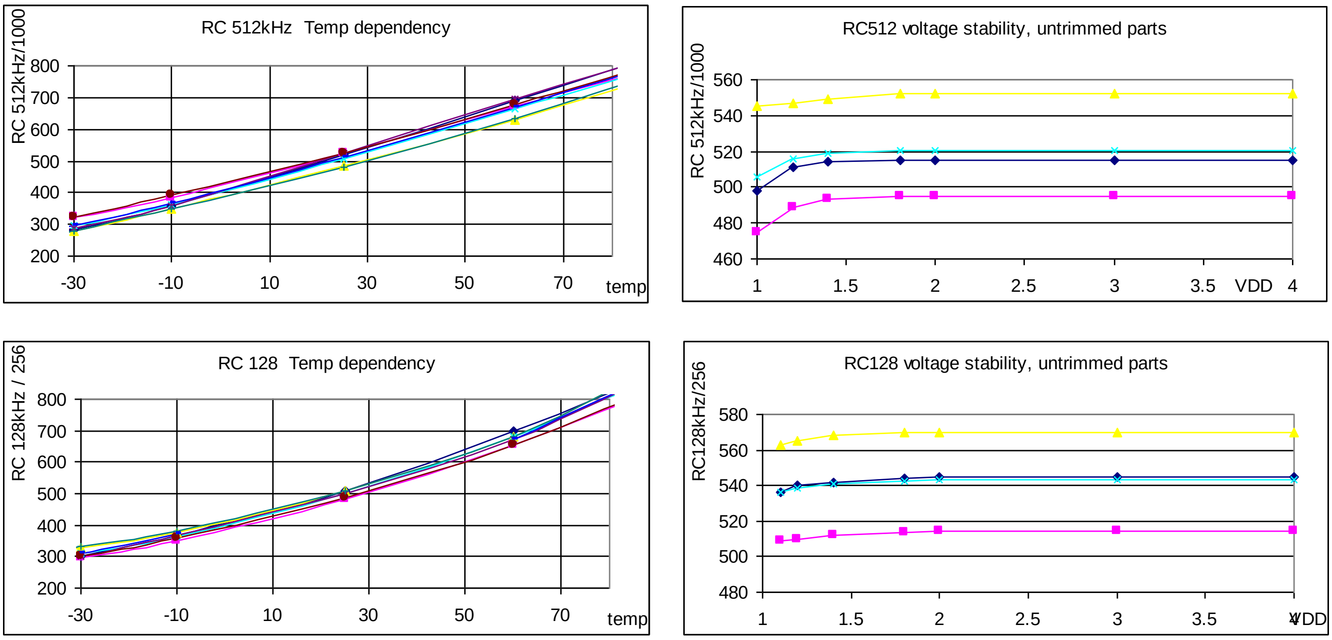
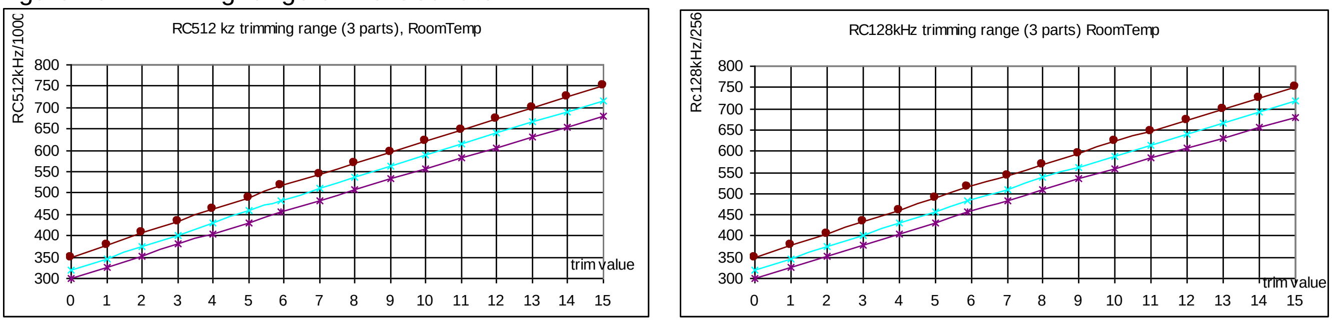


Figure 10. Trimming range of RC Oscillator



5.2 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 the prescaler to Hex(0001).

Table 5.2.1 Prescaler Clock Name Definition

Function	Name	frequency
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

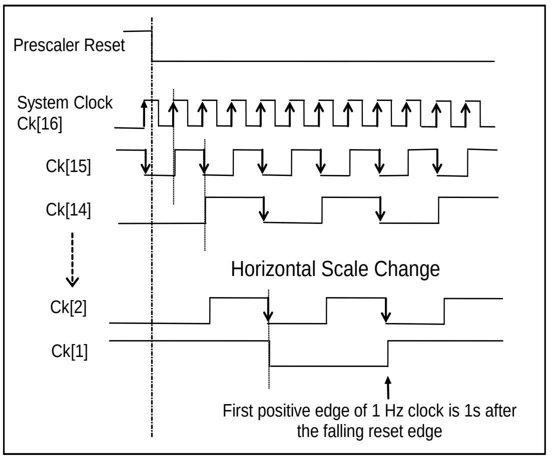
Function	Name	frequency
System clock / 256	Ck[8]	128 Hz
System clock / 512	Ck[7]	64 Hz
System clock / 1024	Ck[6]	32 Hz
System clock / 2048	Ck[5]	16 Hz
System clock / 4096	Ck[4]	8 Hz
System clock / 8192	Ck[3]	4 Hz
System clock / 16384	Ck[2]	2 Hz
System clock / 32768	Ck[1]	1 Hz

Above frequencies are calculated based on a system clock of 32768Hz (RC Osc based)

Table 5.2.2 Control of Prescaler Register RegPresc

B	Name	Reset	R/W	Description
3	**HiDebCkSel**	0	R/W	Hi freq Debouncer clock selection 0 -> Debouncer with Ck[11] 1 -> Debouncer with Ck[14]
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 clock according **HiDebCkSel** bit

Figure 11. Prescaler Frequency Timing



With **DebSel** = 1 one may choose either the Ck[11] or Ck[14] debouncer frequency by selecting bit **HiDebCkSel** in register **RegPresc**. Relative to 32kHz the corresponding max. debouncer times are then 2 ms(Ck11) or 0.25 ms(Ck14). The default debouncer selection is with **DebSel**='0' with a debounce time of 15ms.

Switching the **PrIntSel** may generate an interrupt request. Avoid it with **MaskIRQ32/8** = 0 selection during the switching operation.

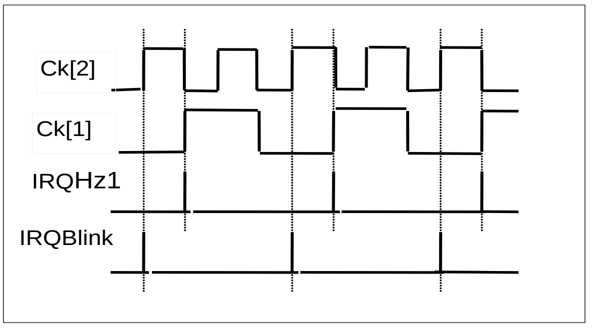
The prescaler contains 3 interrupt sources:
- IRQ32/8 ; this is Ck[6] or Ck[4] positive edge interrupt, the selection is depending on bit **PrIntSel.**
- IRQHz1 ; this is Ck[1] positive edge interrupt
- IRQBlink ; this is 3/4 of Ck[1] period interrupt

There is no interrupt generation on reset.

The first IRQHz1 Interrupt occurs 1 sec (32kHz) after reset.

A possible application for the IRQBlink is LCD-Display blinking control together with IRQHz1.

Figure 12. Prescaler Interrupts



6. Input and Output Ports

The EM6627 has:

- One 4-bit input port (port A)
- One 4-bit input/output port. (port B)
- One serial interface (port SP) also configurable as 4-bit I/O port

Pull-up and pull-down resistors can be added to all this ports with the hardware configuration registers.

6.1 Ports Overview

Table 6.1.1 Input and Output Ports Overview

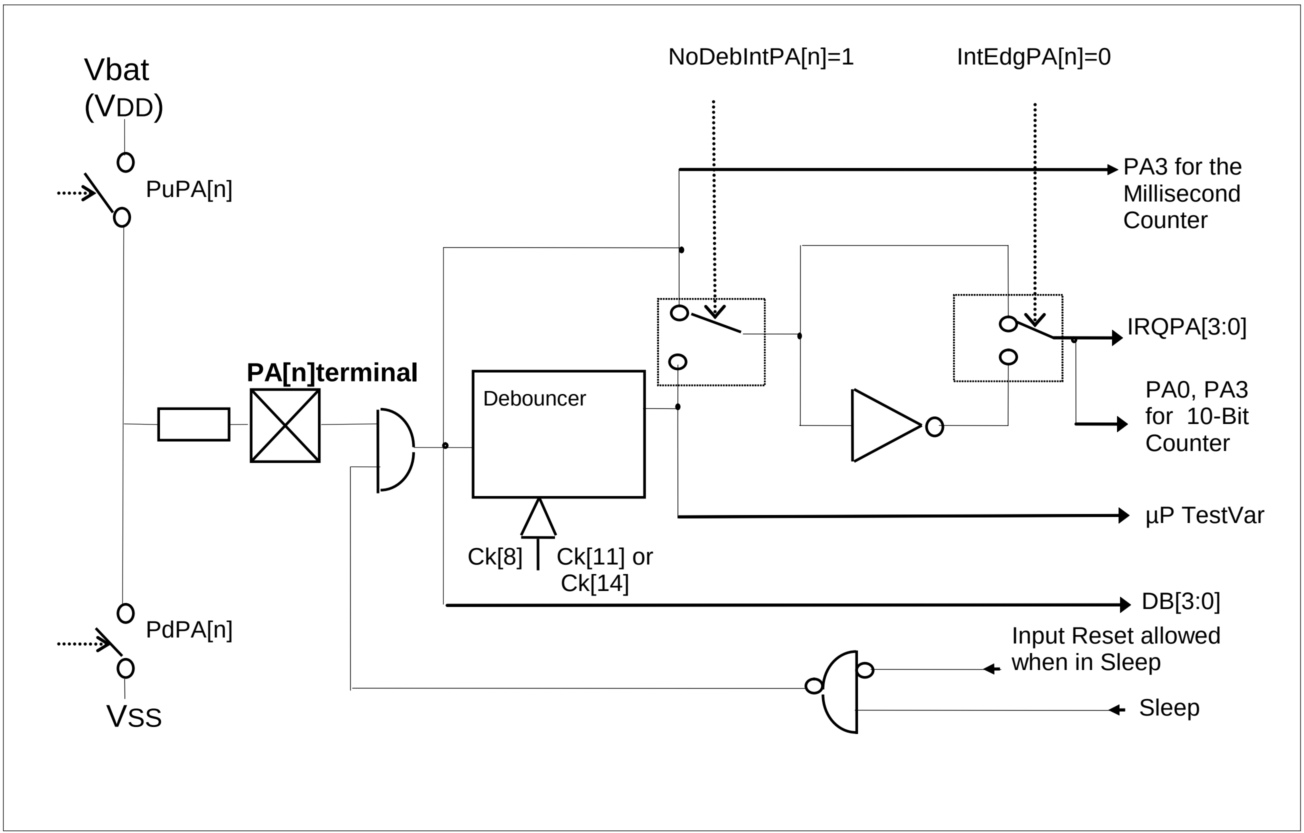
Port	Mode	Register settings	Function	Bit-wise Multifunction on Ports			
PA [3:0]	Input	R: Pull-up R: Pull-down (default) R: Pull(up/down) select R: Debouncer or direct input for IRQ requests and Counter R: + or - for IRQ-edge and counter R: Input reset combination	-Input -Bit-wise interrupt request -Software test variable conditional jump -PA[3],PA[0] input for the event counter -Port A reset inputs	**PA[3]** 10 bit event counter clock start/stop of MSC	**PA[2]** - -	**PA[1]** - -	**PA[0]** 10 bit event counter clock -
PB [3:0]	Individual input or output	R: CMOS or Nch. open drain output R: Pull-down on input R: Pull-up on input	-Input or output -PB[3] for the PWM output -PB[2:0] for the Ck[16,11,1] output -Tristate output	**PB[3]** PWM output	**PB[2]** Ck[16] output	**PB[1]** Ck[11] output	**PB[0]** Ck[1] output
PS [3:0]	Serial I/O or port-wise input / output	R: CMOS or Nch open drain output R: Pull-down on input R: Pull-up on input	-PSP[3], serial clock out -PSP[2], serial data out -PSP[1], serial status out -PSP[0], serial data in -PSP[3:0] 4-bit input/output -Tristate output	**PSP[3]** Serial clock output SCLK	**PSP[2]** Serial data output SOUT	**PSP[1]** Ready or CS Ready/CS	**PSP[0]** Serial data input SIN

6.2 Port A

The EM6627 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 configuration register **CRegFSelPB** 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 EM6627 from sleep mode (all pull resistors remain).

Figure 13. Input Port A Configuration



6.2.1 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 (**CRegDebIntPA**) 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 respectively 0.25 ms depending on the **DebSel** and **HiDebCkSel** bit settings.
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 configuration register **CRegIntEdgPA.**

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 9.
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.

6.2.2 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 configuration register settings. The pull down resistor can be inhibited using the **NoPdPA[n]** bits in the register **CRegNoPdPA.** Instead of Pulldown a pullup resistor can be selected by setting the corresponding **PuPA** bit in register **CRegPuPA**. Pullup has priority over pulldown setting.

Table 6.2.1. Pull-up or Pull-down Resistor on Port A Inputs

NoPdPA[n] value	PuPA[n] value	Action
0	0	pull-down
0	1	pull-up
1	0	no pull-up, no pull-down
1	1	Pull-up

6.2.3 Software Test Variables

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

- Debounced PA[0] is connected to CPU TestVar1.
- Debounced PA[1] is connected to CPU TestVar2.
- Debounced PA[2] is connected to CPU TestVar3.

6.2.4 Port A for 10-Bit Counter and MSC

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 **CRegDebIntPA,** and non-inverted or inverted input with the register **CRegIntEdgPA**. Debouncer input is always recommended.

6.3 Port A Registers

Table 6.3.1 Register RegPA

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

* Direct read on port A terminals

Table 6.3.2 Register RegIRQMask1

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

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

Table 6.3.3 Register RegIRQ1

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

*; Write "1" clears the bit, write "0" has no action, default "0" is: no interrupt request

Table 6.3.4 Register CRegIntEdgPA

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

Default "0" is: Positive edge selection

Table 6.3.5 Register CRegDebIntPA

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

Default "0" is: Debounced inputs for interrupt generation

Table 6.3.6 Register CRegNoPdPA

Bit	Name	power on value	R/W	Description
3	NoPdPA[3]	0	R/W	Pulldown selection on PA[3]
2	NoPdPA[2]	0	R/W	Pulldown selection on PA[2]
1	NoPdPA[1]	0	R/W	Pulldown selection on PA[1]
0	NoPdPA[0]	0	R/W	Pulldown selection on PA[0]

Table 6.3.7 Register CRegPuPA

Bit	Name	power on value	R/W	Description
3	PuPA[3]	0	R/W	Pullupselection on PA[3]
2	PuPA[2]	0	R/W	Pullup selection on PA[2]
1	PuPA[1]	0	R/W	Pullup selection on PA[1]
0	PuPA[0]	0	R/W	Pullup selection on PA[0]

6.4 Port B

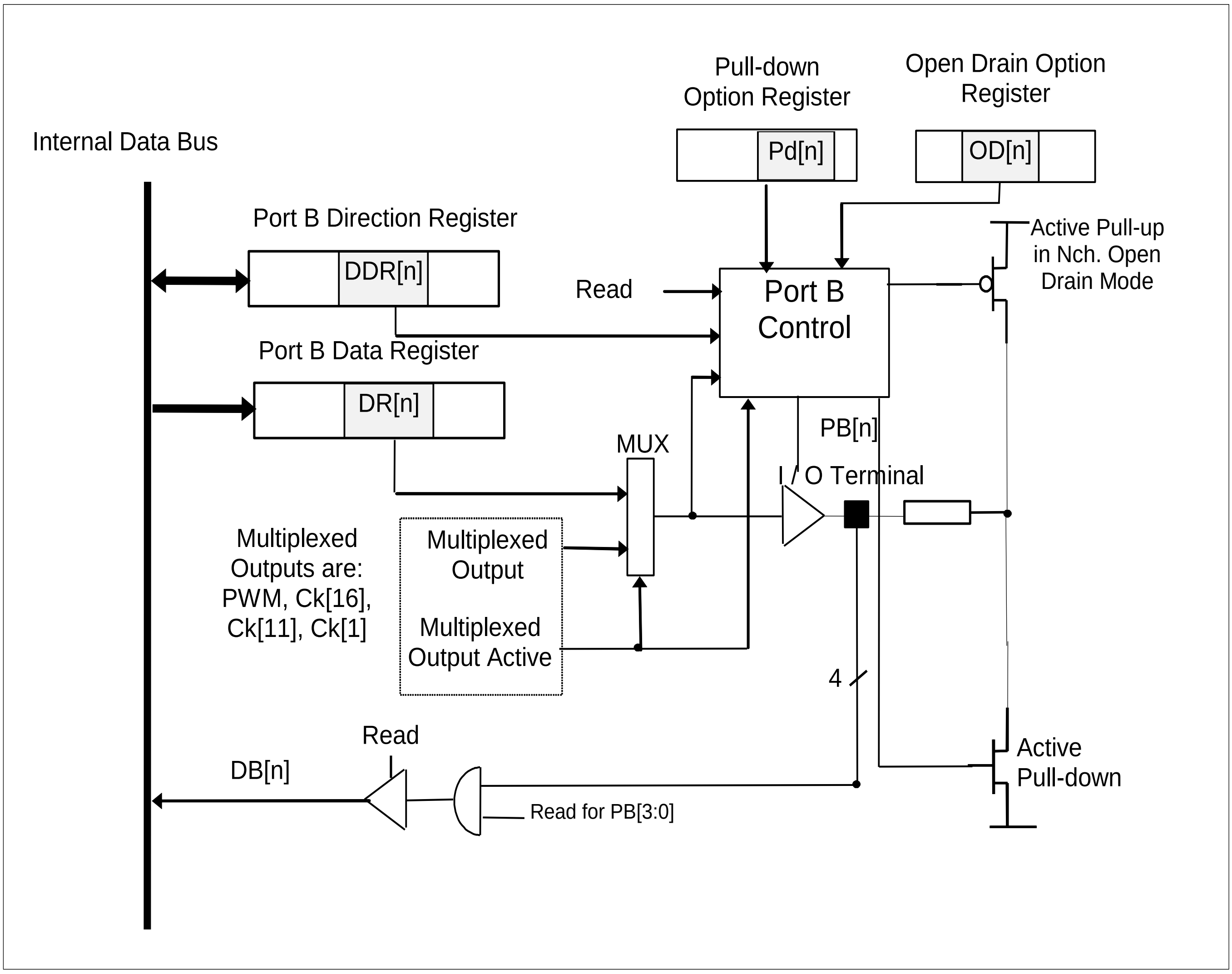
The EM6627 has one four bit general purpose I/O port. Each bit can be configured individually by software for input/output, pull-up, pull-down and CMOS or Nch. open drain output type. The port outputs either data, frequency or PWM signals.

6.4.1 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 14 for details.

Figure 14. Port B Architecture



6.4.2 Pull-up or Pull-down

On each terminal of PB[3:0] an internal input pull-up or pull-down resistor can be connected per hardware configuration register.

Pull-down ON : bit **NoPdPB**[n] must be '0' and **NchOpDPB**[n] = '0'
(Pullup has priority over Pulldown)

Pull-down OFF: **NoPdPB**[n] = '1' cuts off the pull-down.

Pull-up ON * : 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* : **NchOpDPB**[n] = '0' cuts off the pull-up,
OR if **NchOpDPB**[n] = '1' then **PBData**[n] = 0 cuts off the pull-up.

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

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 :

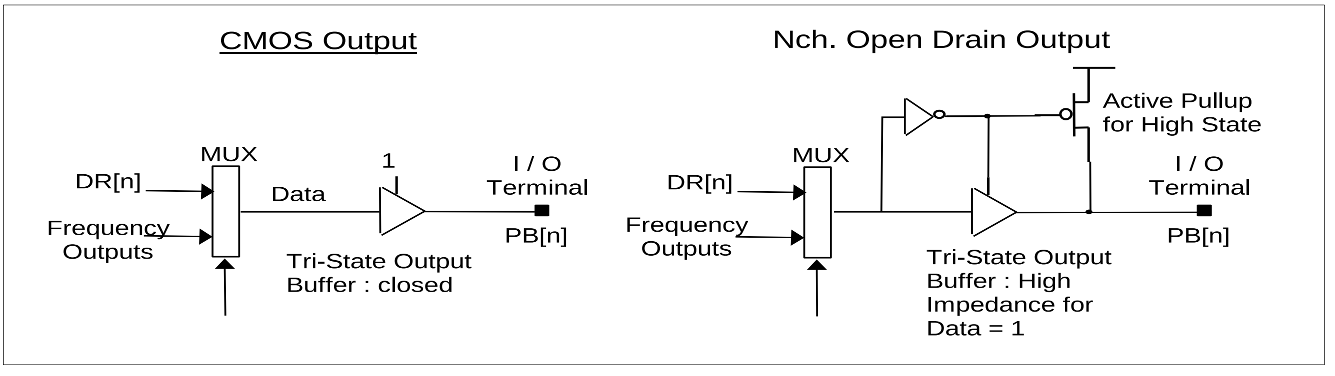
- Switch on the pull resistor.
- Allow sufficient time - RC constant - for the pull resistor to drive the line to either VSS or VDD.
- Read the port B
- Switch off the pull resistor

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.

6.4.3 CMOS / NCH. Open Drain 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 15. CMOS or Nch. Open Drain Outputs



6.4.4 PWM and Frequency Output

PB[3] can also be used to output the PWM (Pulse Width Modulation) signal from the 10-Bit Counter, the Ck[16], Ck[11] as well as the Ck[1] prescaler frequencies.

-Selecting PWM output on PB[3] with bit **PWMOn** in register **RegSysCntl1** and running the counter.
-Selecting Ck[16] output on PB[2] with bit **PB32kHzOut** in register **CRegFSelPB**
-Selecting Ck[11] output on PB[1] with bit **PB1kHzOut** in register **CRegFSelPB**
-Selecting Ck[1] output on PB[0] with bit **PB1HzOut** in register **CRegFSelPB**

6.5 Port B Registers

Table 6.5.1 Register RegPBData

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

* : Direct read on the port B terminal (not the internal register)

Table 6.5.2 Register RegPBCntl

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

Default "0" is: port B in input mode

Table 6.5.3 Configuration Register CRegFSelPB

Bit	Name	power on value	R/W	Description
3	*InpResSleep*	*0*	*R/W*	*Reset from sleep with port A*
2	PB32kHzOut	0	R/W	Ck[16] output on PB[2]
1	PB1kHzOut	0	R/W	Ck[11] output on PB[1]
0	PB1HzOut	0	R/W	Ck[1] output on PB[0]

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

Table 6.5.4 Configuration Register CRegNoPdPB

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

Default "0" is: Pull-down on

Table 6.5.5 Configuration Register CRegNchOpDPB

Bit	Name	power on value	R/W	Description
3	NchOpDPB[3]	0	R/W	Nch. Open Drain on PB[3]
2	NchOpDPB[2]	0	R/W	Nch. Open Drain on PB[2]
1	NchOpDPB[1]	0	R/W	Nch. Open Drain on PB[1]
0	NchOpDPB[0]	0	R/W	Nch. Open Drain on PB[0]

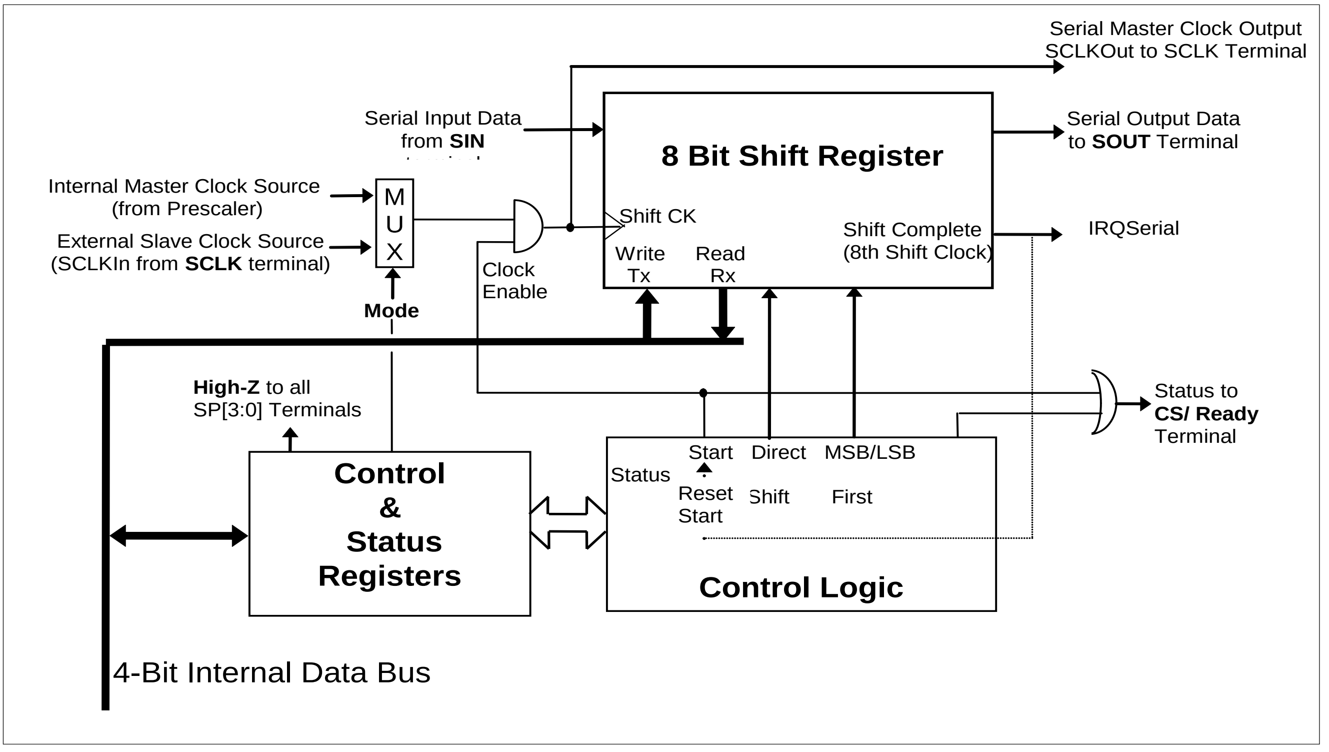
Default "0" is: CMOS output

6.6 Port Serial

The EM6627 contains a simple, half duplex three wire synchronous type serial interface., which can be used to program or read an external EEPROM, ADC, ... etc.

For data reception, a shift-register converts the serial input data on the SIN(PSP[0]) terminal to a parallel format, which is subsequently read by the CPU in registers **RegSDataL** and **RegSDataH** for low and high nibble. To transmit data, the CPU loads data into the shift register, which then serializes it on the SOUT(PSP[2]) terminal. It is possible for the shift register to simultaneously shift data out on the SOUT terminal and shift data on the SIN terminal. In Master mode, the shifting clock is supplied internally by the Prescaler : one of three prescaler frequencies are available, Ck[16], Ck[15] or Ck[14]. In Slave mode, the shifting clock is supplied externally on the SCLKIn(PSP[3]) terminal. In either mode, it is possible to program : the shifting edge, shift MSB first or LSB first and direct shift output. All these selection are done in register **RegSCntl1** and **RegSCntl2**.

Figure 16. Serial Interface Architecture



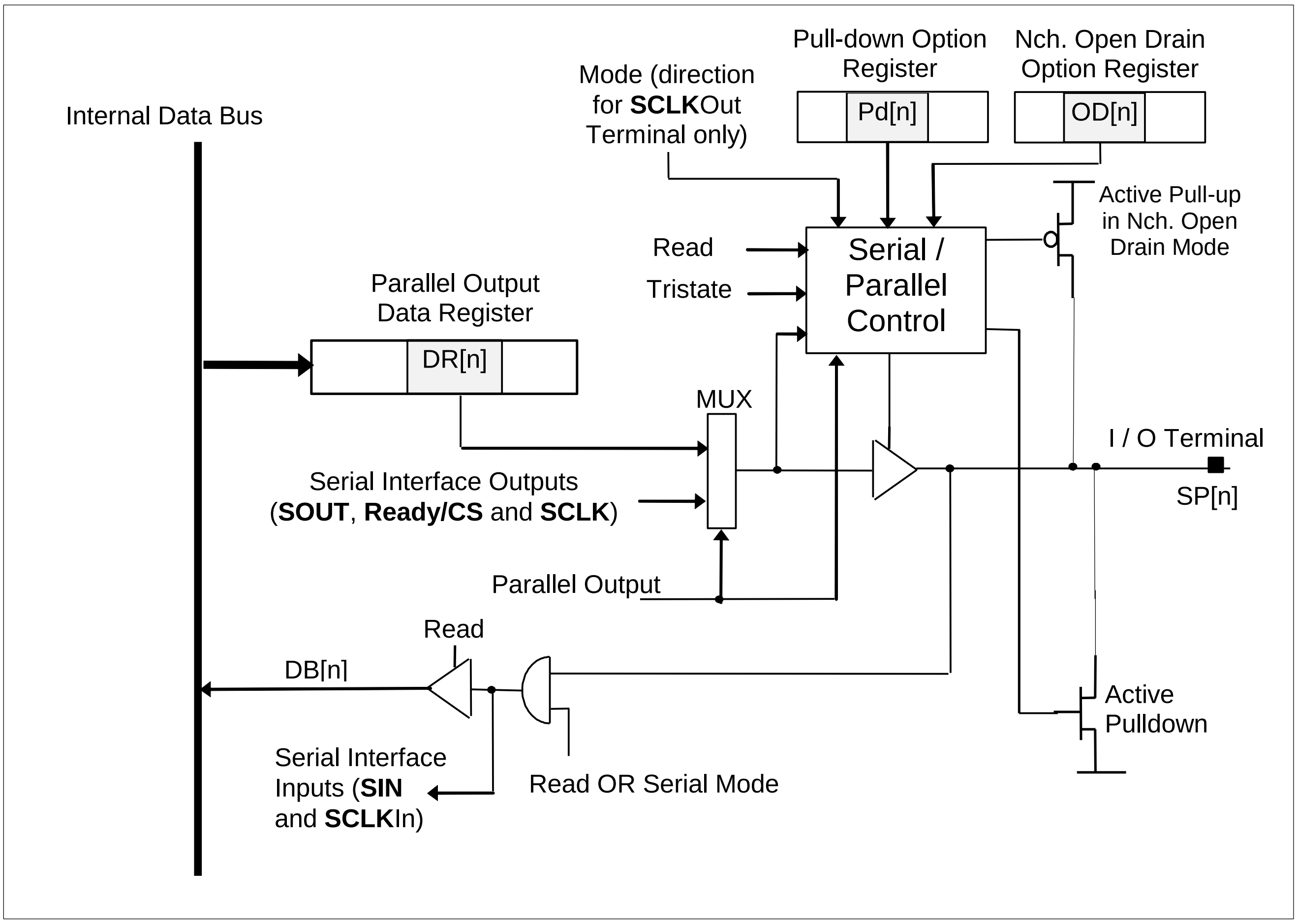
The PSP[3..0] terminal configuration is shown in Figure 17. When the Serial Interface is active then :

* PSP[1] {Ready / CS) is outputting the ready (slave mode) or the CS signal (master mode).
* PSP[2] {SOUT} is always an output.
* PSP[0] {SIN} is always an input.
* PSP[3] {SCLK} is an output for Master mode {SCLKOut} and an input for Slave mode {SCLKIn}

6.6.1 4-bit Parallel I/O

Selecting **OM[1],OM[0]** = ‘1’ in register **RegSCntl2** the PSP[3:0] terminals are configured as a 4-bit Output. Output data is stored in the register **RegSPData** .
The **RegSPData** is defined as a read/write register, but what is read is not the register output, but the port PSP[3:0] terminal values
Selecting **OM[1],OM[0]** = ‘0’ in register **RegSCntl2** the PSP[3:0] outputs are cut off (tristate). The terminals can be used as inputs with individual (bit-wise) pull-up or pull-down settings.
Independent of the selected configuration, the PSP[3:0] terminal levels are always readable.

Figure 17. Port SP Terminal Configuration



6.6.2 Pull-up or Pull-down

On each terminal of PPS[3:0] an internal input pull-up or pull-down resistor can be connected per hardware configuration register.

Pull-down ON : bit **NoPdPS**[n] must be '0' and **NchOpDPS**[n] = '0'
(Pullup has priority over Pulldown)

Pull-down OFF: **NoPdPS**[n] = '1' cuts off the pull-down.

Pull-up ON * : bit **NchOpDPS**[n] must be '1' ,
AND (bit **PSIOCntl[n]** = '0' (input mode) **OR** if **PSIOCntl[n]** = '1' while **PSData**[n] = 1.)

Pull-up OFF* : **NchOpDPS**[n] = '0' cuts off the pull-up,
OR if **NchOpDPS**[n] = '1' then **PSData**[n] = 0 cuts off the pull-up.

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

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

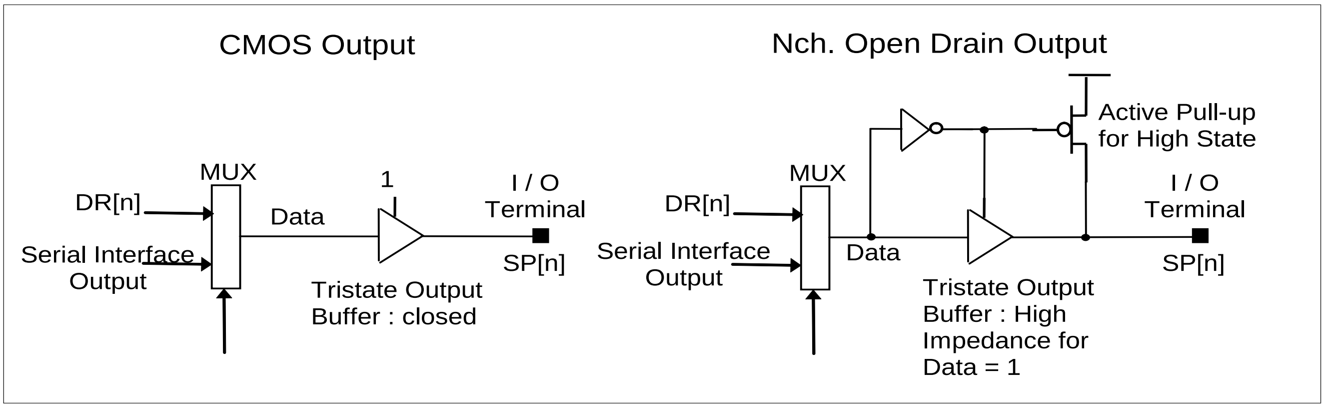
- Switch on the pull resistor.
- Allow sufficient time - RC constant - for the pull resistor to drive the line to either VSS or VDD.
- Read the port PS
- Switch off the pull resistor

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.

6.6.3 Nch. Open Drain Outputs

The port SP 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 out on the terminal, the logic '1' value is high impedance or defined by the internal pull-up resistor (if existing).

Figure 18. CMOS or Nch. Open Drain outputs



6.6.4 General Functional Description

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

Parallel to serial conversion procedure (master mode example).
Write to **RegSCntl1** serial control (clock freq. in master mode, edge and MSB/LSB select).
Write to **RegSDataL** and **RegSDataH** (shift out data values).
Write to **RegSCntl2** (Start=1, mode select, status). ---> Starts the shift out
After the eighth clock an interrupt is generated, **Start** becomes low. Then, interrupt handling

Serial to parallel conversion procedure (slave mode example).
Write to **RegSCntl1** (slave mode, edge and MSB/LSB select).
Write to **RegSCntl2** (Start=1, mode select, status).
After eight serial clocks an interrupt is generated, **Start** becomes low.
Interrupt handling.
Shift register **RegSDataL** and **RegSDataH** read.
A new shift operation can be authorized.

6.6.5 Detailed Functional Description

Master or Slave mode is selected in the control register **RegSCntl1.**
In Slave mode, the serial clock comes from an external device and is input via the PSP[3] terminal as a synchronous clock (SCLKIn) to the serial interface. The serial clock is ignored as long as the **Start** bit is not set. After setting **Start,** only the eight following active edges of the serial clock input PSP[3] are used to shift the serial data in and out. After eight serial clock edges the **Start** bit is reset. The PSP[1] terminal is a copy of the (**Start OR Status)** bit values, it can be used to indicate to the external master, that the interface is ready to operate or it can be used as a chip select signal in case of an external slave.
In Master mode, the synchronous serial clock is generated internally from the system clock. The frequency is selected from one out of three sources (**MS0** and **MS1** bits in **RegSCntl1**) **.** The serial shifting clock is only generated during **Start** = high and is output to the SCLK terminal as the Master Clock (SCLKOut). When **Start** is low, the serial clock output on PSP[3] is 0.
An interrupt request **IRQSerial** is generated after the eight shift operations are done. This signal is set by the last negative edge of the serial interface clock on PSP[3] (master or slave mode) and is reset to 0 by the next write of **Start** or by any reset. This interrupt can be masked with register **RegIRQMask3.** For more details about the interrupt handling see chapter 9.
Serial data input on PSP[0] is sampled by the positive or negative serial shifting clock edge, as selected by the Control Register **POSnNeg** bit**.** Serial data input is shifted in LSB first or MSB first, as selected by the Control Register **MSBnLSB** bit**.**

6.6.6 Output Modes

Serial data output is given out in two different ways (Refer also to Figure 19 and Figure 20).

- **OM[1]** = 1, **OM[0]** = 0 :
The serial output data is generated with the selected shift register clock (**POSnNeg**). The first data bit is available directly after the **Start** bit is set.

-**OM[1]** = 0, **OM[0]** = 1 :
The serial output data is re-synchronized by the positive serial interface clock edge, independent of the selected clock shifting edge. The first data bit is available on the first positive serial interface clock edge after Start='1'.

Figure 19. Direct or Re-Synchronized Output

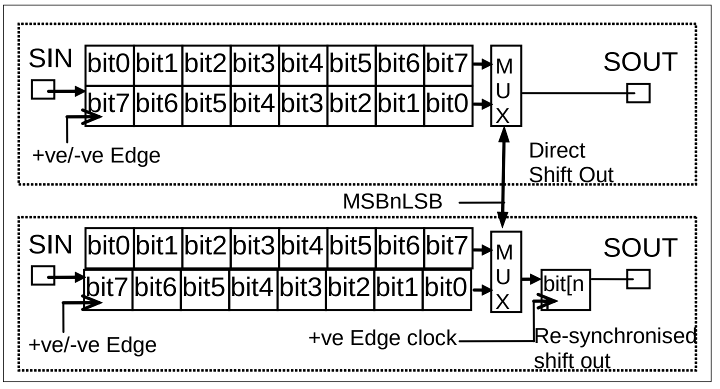
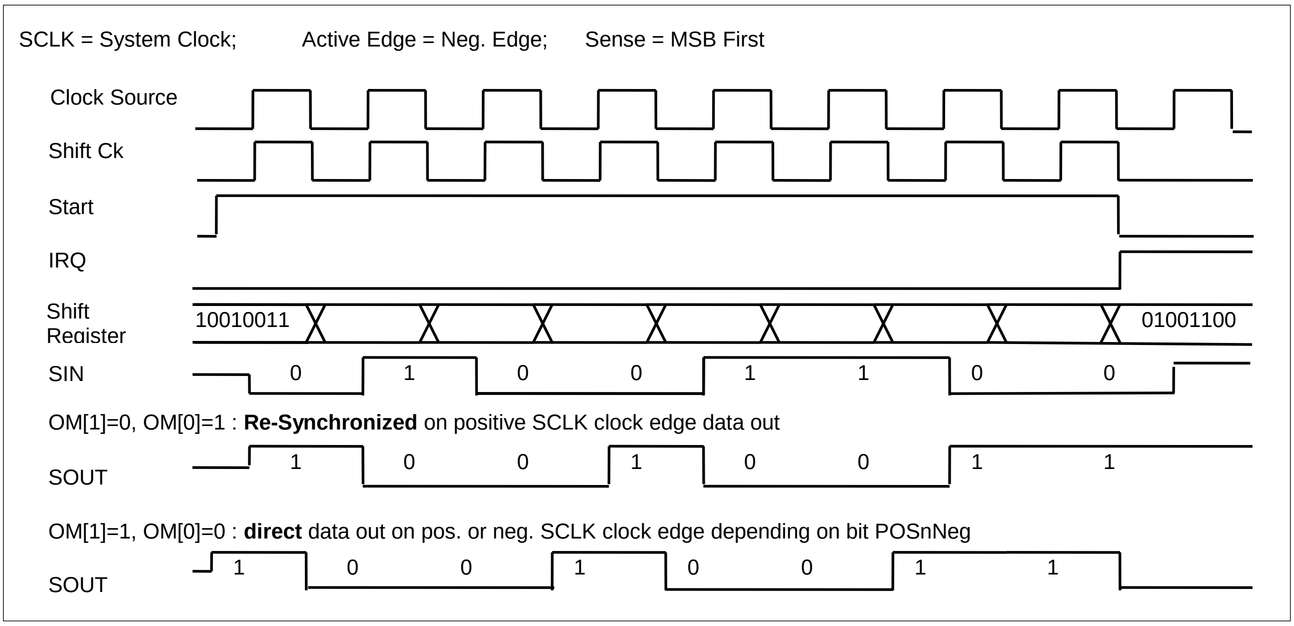


Table 6.6.6 Output Mode Selection in RegSCntl2

OM[1]	OM[0]	Output mode	Description
0	0	Tristate	Output disable (tristate on PSP[3:0])
0	1	Serial-Synchronized	Re-synchronized positive edge data shift out
1	0	Serial-Direct	Direct shift pos. or neg. edge data out
1	1	Parallel	Parallel port SP output

Tristate output is selected by default.

Figure 20 Shift Operation and IRQ Generation

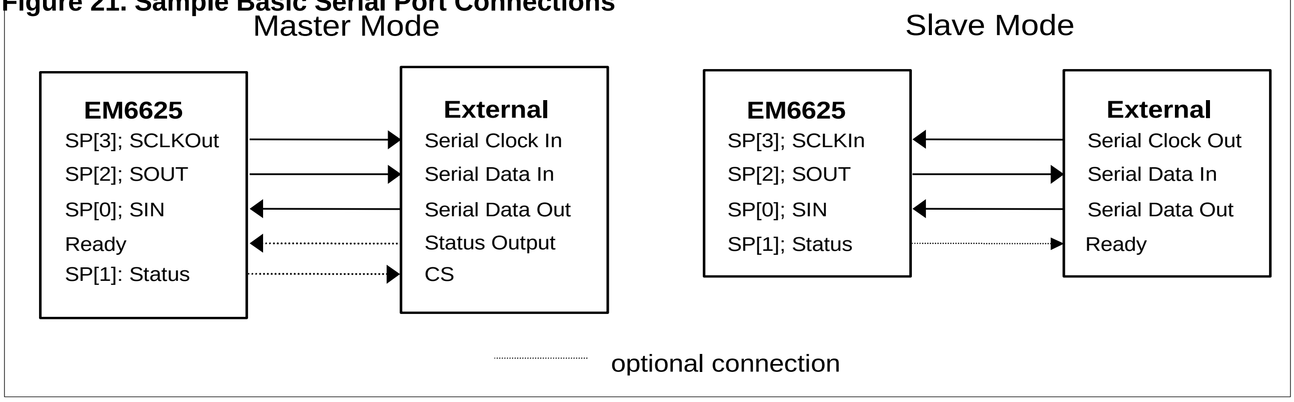


6.6.7 Reset and Sleep on Port SP

During circuit initialization, all hardware configuration registers are reset by Power On Reset and therefore all pull-ups are off and all pull-downs are on. During Sleep mode, Port SP inputs are cut-off , the circuit is in Reset State. However the Reset State does not reset the hardware configuration registers and pull-downs, if previously turned on, remain on even during Sleep mode. After any reset the serial interface parameters are reset to : Slave mode, Start and Status = 0, LSB first, negative edge shift , PSP[3:0] tristate.

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

Figure 21. Sample Basic Serial Port Connections



6.7 Serial Interface Registers

Table 6.7.1 Register RegSCntl1

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

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

Table 6.7.2 Frequency and Master Slave Mode Selection

MS1	MS0	Description
0	0	Slave mode: Clock from external
0	1	Master mode: System clock / 4
1	0	Master mode: System clock / 2
1	1	Master mode: System clock

Table 6.7.3 Register RegSCntl2

Bit	Name	Reset	R/W	Description
3	Start	0	R/W	Enabling the interface,
2	Status	0	R/W	Ready or Chip Select output on PSP[1]
1	OM[1]	0	R/W	Output mode select 1
0	OM[0]	0	R/W	Output mode select 0

Default "0" is: Interface disabled, status 0, serial mode, output tristate.

Table 6.7.4 Register RegSDataL

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

Default "0" is: Data equal 0.

Table 6.7.5 Register RegSDataH

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

Default "0" is: Data equal 0.

Table 6.7.6 Register RegSPData

Bit	Name	Reset	R/W	Description
3	SerPData[3]	0	R* /W	Parallel output data
2	SerPData[2]	0	R* /W	Parallel output data
1	SerPData[1]	0	R* /W	Parallel output data
0	SerPData[0]	0	R* /W	Parallel output data

R* : The input terminal value is read, not the register

Table 6.7.7 Configuration Register CRegNoPdPS

Bit	Name		R/W	Description
3	NoPdPS[3]	0	R/W	No pull-down on PSP[3]
2	NoPdPS[2]	0	R/W	No pull-down on PSP[2]
1	NoPdPS[1]	0	R/W	No pull-down on PSP[1]
0	NoPdPS[0]	0	R/W	No pull-down on PSP[0]

Default "0" is: Pull-down on

Table 6.7.8 Configuration Register CRegNchOpDPS

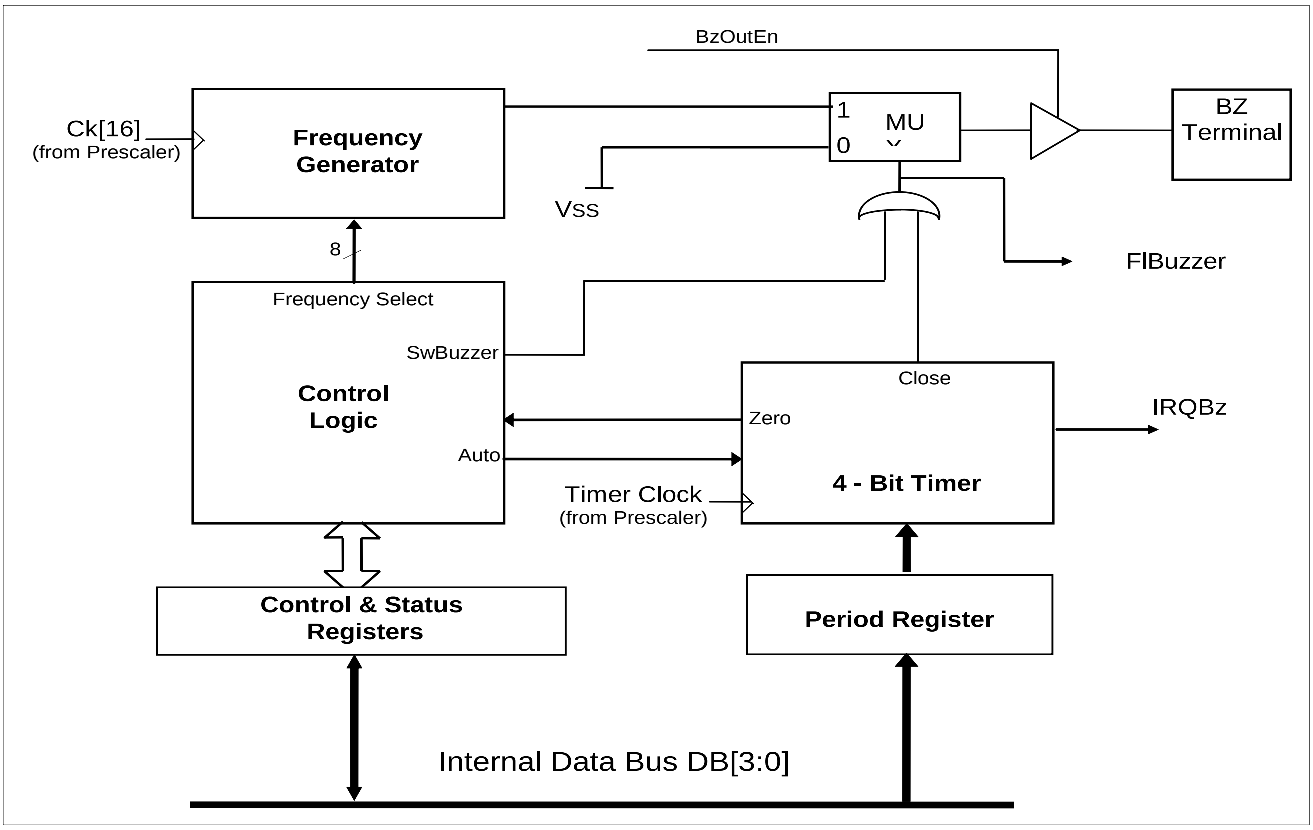
Bit	Name		R/W	Description
3	NchOpDPS[3]	0	R/W	Nch. Open Drain on PSP[3]
2	NchOpDPS[2]	0	R/W	Nch. Open Drain on PSP[2]
1	NchOpDPS[1]	0	R/W	Nch. Open Drain on PSP[1]
0	NchOpDPS[0]	0	R/W	Nch. Open Drain on PSP[0]

Default "0" is: CMOS output

7. Melody, Buzzer

A normal application is to drive a buzzer connected onto the terminal Buzzer.
This peripheral cell is a combination of a 7 frequency tone generator and a 4-bit timer, used to provide a 50% duty cycle signal on the Buzzer terminal of a pre-selected length and frequency. The Buzzer terminal is active as long as the timer is not 0 or the **SwBuzzer** is set to '1'. The 4-bit timer can be used for another application independent of the Buzzer terminal by selecting "silence" instead of another frequency on the Buzzer output. "Silence" can also be used as part of a melody, or to switch off the buzzer.
To use the buzzer independent of the 4-bit timer one has to set the switch **SwBuzzer.** This bit is in register **RegMelTim** and selects the signal duration on the buzzer output. If **SwBuzzer**=1 then the signal is output until the bit is set back to 0 . With **SwBuzzer**=0 the output signal duration is controlled by the 4bit timer. If neither the **SwBuzzer** or the timer are active, the Buzzer terminal is on 0.
The high impedance state setting with **BzOutEn** is independent of the **SwBuzzer** and Timer settings. As soon as the bit is set to 1 the Buzzer terminal is set tristate. See also Figure 22.

Figure 22. Melody Generator Block Diagram



7.1 4-Bit Timer

The timer has 2 modes:

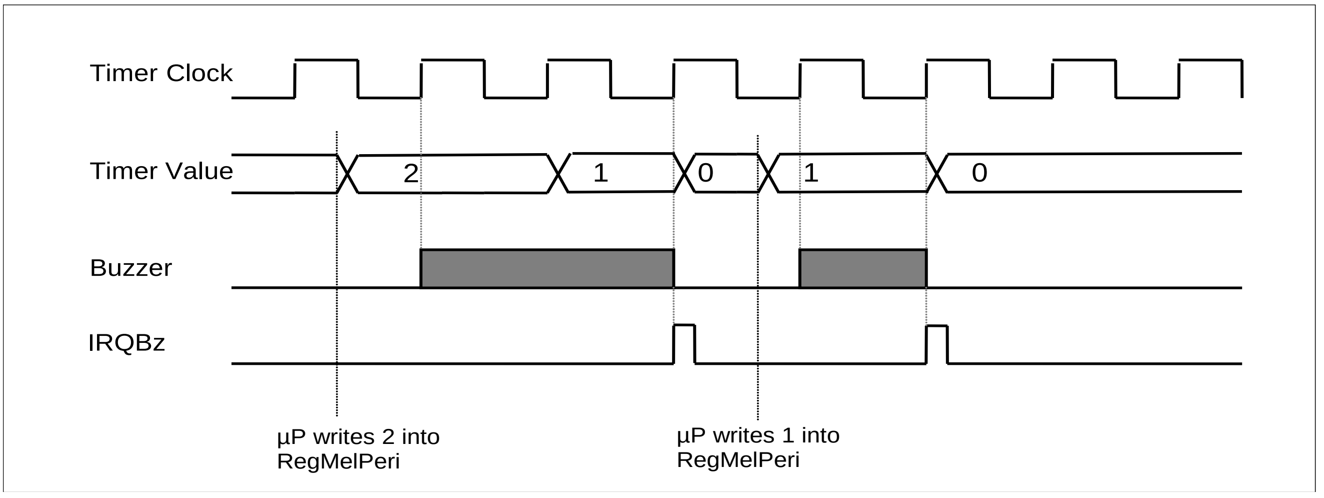
- Single run mode (**Auto**=0)
- Continuos run mode (**Auto**=1)

Mode selection and timer count down frequency is done in register **RegMelTim.** All timer frequencies are coming from the prescaler. The 4-bit timer can be used independent of the melody buzzer application.
Whenever the timer reaches 0 it generates an interrupt request **IRQBz** in the register **RegIRQ2** . This interrupt can be masked with the bit **MaskIRQBz** in register **RegIRQMask2.** By writing 0 into the timer period register the timer stops immediately and does not generate an interrupt.

7.1.1 Single Run Mode

The RegMelPeri value and the selected timer frequency in RegMelTim control the timer duration. The timer is counting down from its previously charged value until it reaches 0. On 0 the timer stops and generates an interrupt request. The buzzer frequency output is enabled after the next positive timer clock edge and remains enabled until the timer reaches 0.

Figure 23. Single Run Mode

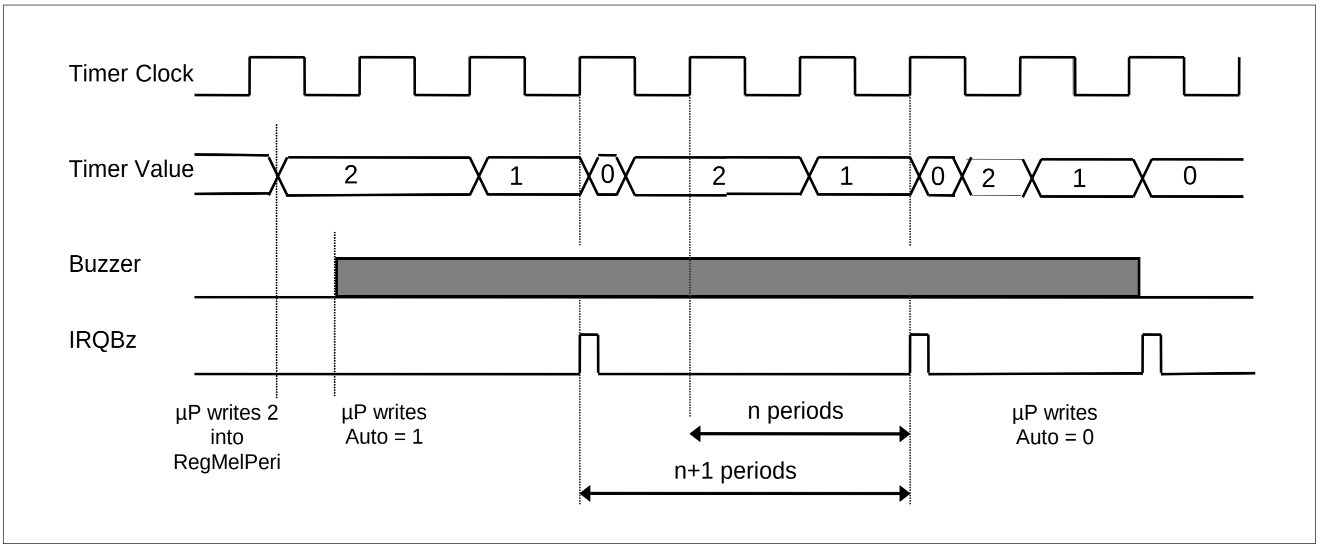


7.1.2 Continuos Run Mode

This is almost the same as the single run mode only that in this case the timer after reaching 0 reloads itself automatically with the register **RegMelPeri** value. Every time the timer reaches 0 an interrupt request is send. There are 2 ways to stop the continuos mode.

- First, changing the mode to single run mode. As the timer reaches 0 it stops. The last period after **Auto**=0 is of length **RegMelPeri + 1**.
- Second, loading 0 into the timer period register **RegMelPeri** stops the timer immediately, no interrupt is generated and the **Auto** flag is reset. The buzzer frequency output is enabled directly by writing **Auto**=1.

Figure 24. Continuos Run Mode



7.2 Programming Order

Single run mode usage

1st, selecting the buzzer frequency into **RegMelFSel**.
2nd, selecting the timer clock frequency in **RegMelTim**.
3rd, selecting the timer period in **RegMelPeri**.
--> On the next positive clock edge the buzzer output is enabled.

Continuos run mode usage

1st, selecting the buzzer frequency into **RegMelFSel**.
2nd, selecting the timer clock frequency in **RegMelTim** (**Auto**=0).
3rd, selecting the timer period in **RegMelPeri**.
4th, set bit **Auto** in **RegMelTim**.
--> Immediately the buzzer output is active.

Avoid timer clock frequency switch during buzzer operation.

7.3 Melody Registers

Table 7.3.1 Register RegMelFSel

Bit	Name	Reset	R/W	Description
3	BzOutEn	0	R/W	Buzzer Output tristate
2	MelFSel[2]	0	R/W	Buzzer frequency select
1	MelFSel[1]	0	R/W	Buzzer frequency select
0	MelFSel[0]	0	R/W	Buzzer frequency select

Default : Buzzer tristate, silence

Table 7.3.2 Buzzer Output Frequency Selection with MelFSel[2..0]

MelFSel[2]	MelFSel[1]	MelFSel[0]	Frequency	
0	0	0	Vss (silence)	
0	0	1	SysClock/8	DO8
0	1	0	SysClock/10	SOL7#
0	1	1	SysClock/12	FA7
1	0	0	SysClock/14	RE7
1	0	1	SysClock/16	DO7
1	1	0	SysClock/20	SOL6#
1	1	1	SysClock/24	FA6

Table 7.3.3 Register RegMelTim

Bit	Name	Reset	R/W	Description
3	SwBuzzer FlBuzzer	0 0	W R	Write: switch buzzer Read: flag buzzer
2	Auto	0	R/W	Single or continus run mode
1	FTimSel1	0	R/W	Timer clock frequency select
0	FTimSel0	0	R/W	Timer clock frequency select

Default : Single run mode, Ck[3] from prescaler as timer clock

Table 7.3.4 Timer Clock Frequency Select

FTimSel0	FTimSel1	Timer Clock	On 32 KHz operation
0	0	Ck[3]	4 Hz
1	0	Ck[5]	16 Hz
0	1	Ck[7]	64 Hz
1	1	Ck[1]	1 Hz

Table 7.3.5 Register RegMelPeri

Bit	Name	Reset	R/W	Description
3	Per[3]	0	W	Melody timer period MSB
2	Per[2]	0	W	Melody timer period
1	Per[1]	0	W	Melody timer period
0	Per[0]	0	W	Melody timer period LSB

The total timer period duration is calculated as following:

Duration = Value(RegMelPeri) x 1/Ck[n]

Where, Ck[n] is the timer clock frequency and Value(RegMelPeri) is the value of the register **RegMelPeri.**

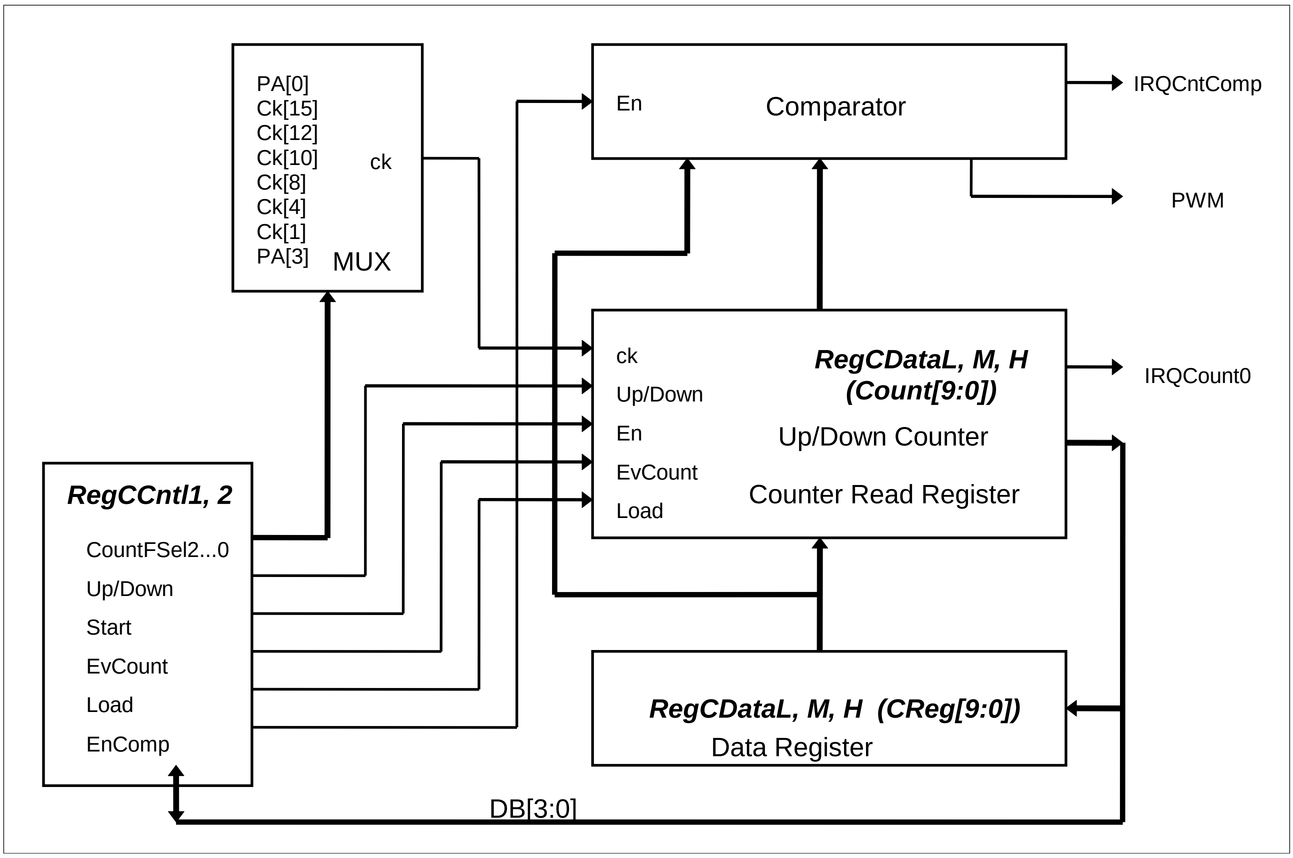
8. 10-bit Counter

The EM6627 has a built-in universal cyclic counter. It can be configured as 10, 8, 6 or 4-bit counter. If 10-bits are selected we call that full bit counting, if 8, 6 or 4-bits are selected we call that limited bit counting.
The counter works in up- or down count mode. Eight clocks can be used as the input clock source, six of them are prescaler frequencies and two are coming from the input pads PA[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 9 for more information about the interrupt handling.

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

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

Figure 25. 10-bit Counter Block Diagram



8.1 Full and Limited Bit Counting

Table 7.3.1. 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

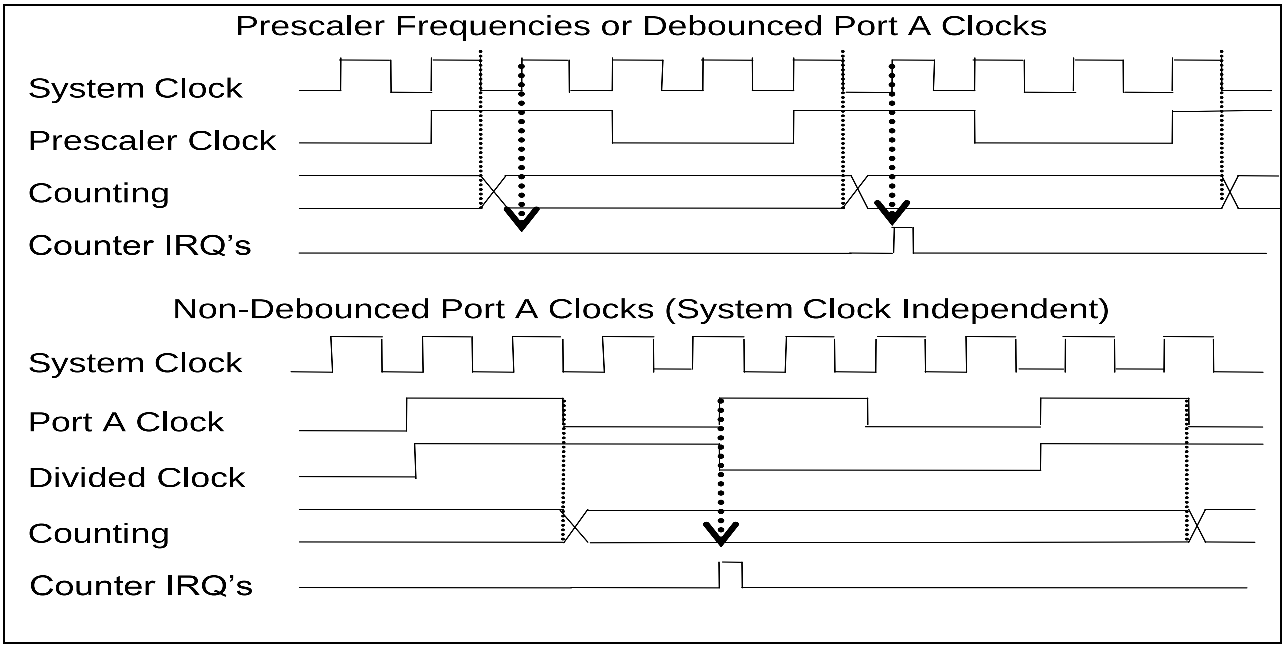
In Full Bit Counting mode the counter uses its maximum of 10-bits length (default). With the **BitSel[1,0]** bits in register **RegCDataH** one can lower the counter length, for IRQ generation, to 8, 6 or 4 bits. This means that actually the counter always uses all the 10-bits, but IRQCount0 generation 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.

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 26. Counter Clock Timing



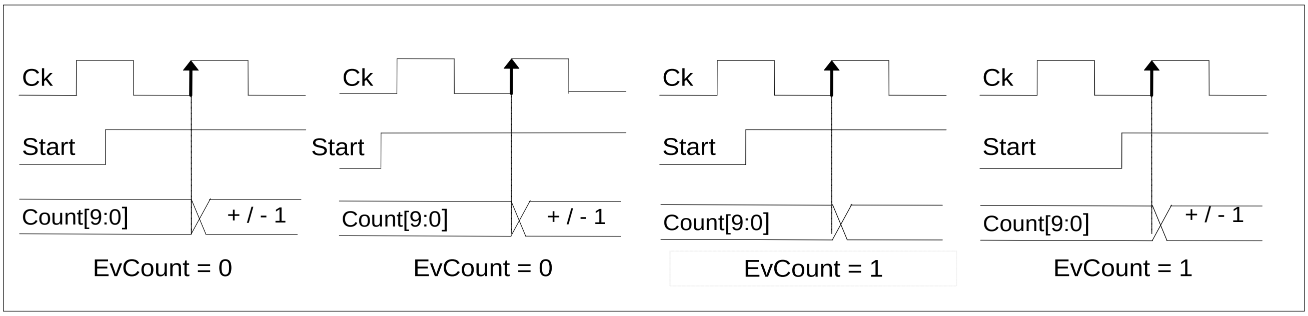
The two remaining clock sources are coming from the PA[0] or PA[3] terminals. Refer to the Figure 13 on page 16 for details. Both sources can be either debounced (Ck[11] or Ck[8]) or direct inputs, the input polarity can also be chosen. The output after the debouncer polarity selector is named PA3 , PA0 respectively. For the debouncer and input polarity selection refer to chapter 6.3.

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 EM6627 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 (@$V_{DD} \geq 1.5$ V). 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 27. 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 **CRegIntEdgPA** and should be debounced. The debouncer is switched on in register **CRegDebIntPA** 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 13 for internal clock signal generation.

Figure 27. 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 **RegSysCntl1** routes the counter PWM output to port B terminal PB[3]. Insure that PB[3] is set to output mode . Refer to section 6.4 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 28. PWM Output in Up Count Mode

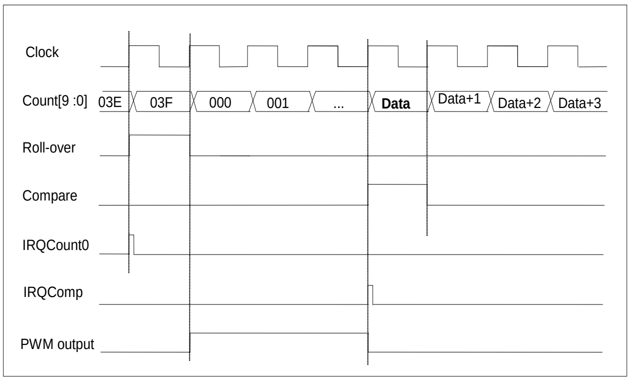
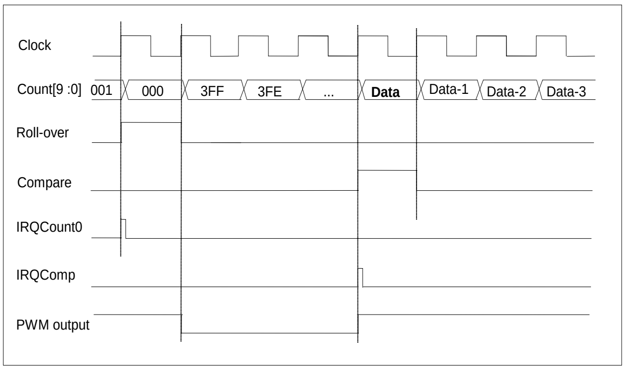


Figure 29. PWM Output in Down Count Mode



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 count[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.
The default count[9:0] value is 0x3FF in downcount and 0x000 in upcount mode.

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** 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	*PWMOn*	*0*	*R/W*	*PWM output on PB[3]*

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 (Up), 1(Down)	R	Data register bit 7, updated at read of RegCDataL
2	ShCount[6]	0 (Up), 1(Down)	R	Data register bit 6, updated at read of RegCDataL
1	ShCount[5]	0 (Up), 1(Down)	R	Data register bit 5, updated at read of RegCDataL
0	ShCount[4]	0 (Up), 1(Down)	R	Data register bit 4, updated at read of RegCDataL

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 (Up), 1(Down)	R	Data register bit 9, updated at read of RegCDataL
0	ShCount[8]	0 (Up), 1(Down)	R	Data register bit 8, updated at read of RegCDataL

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

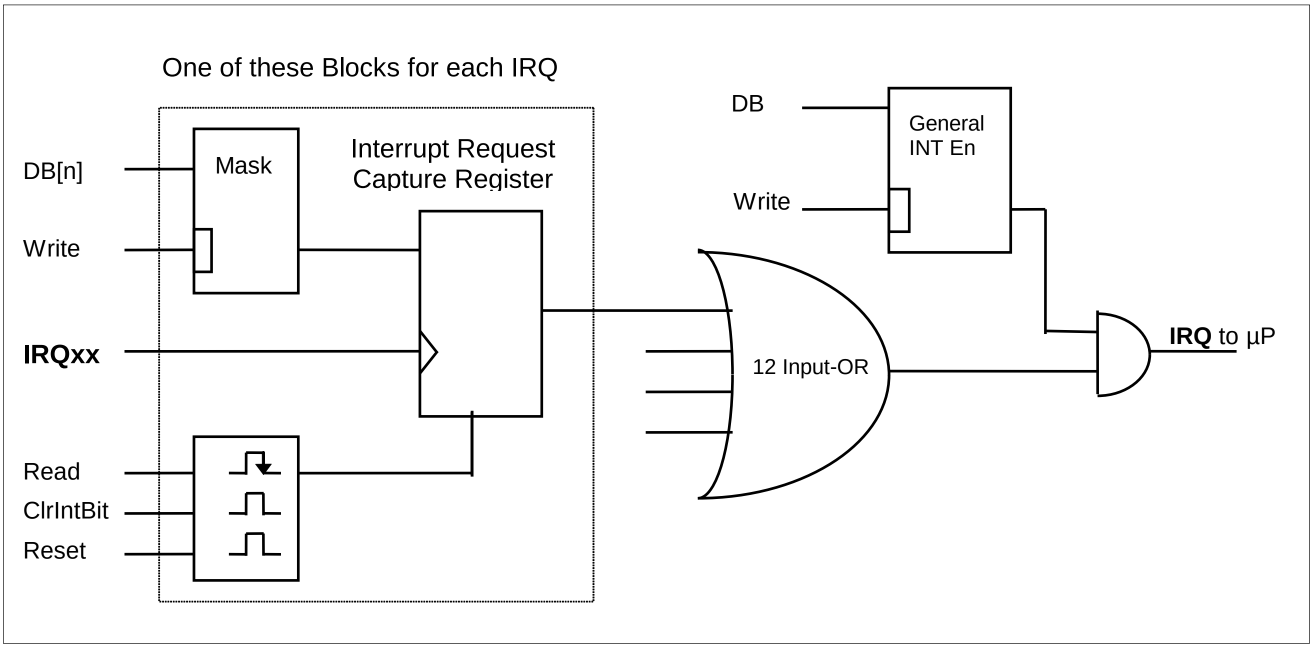
9. Interrupt Controller

The EM6627 has 12 different interrupt request sources, each of which is maskable. Five of them come from external sources and seven from internal sources.

External(4)	**- Port A,**	**PA[3] .. PA[0] inputs**
	- Serial Interface	
Internal(7)	**- Prescaler**	**Ck[1], Blink, 32Hz/8Hz**
	- Melody timer	
	- Serial Interface	
	- 10-bit Counter	**Count0, CountComp**
	- EEPROM end of write	

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 30. Interrupt Controller Block Diagram



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

After each read operation on the interrupt request registers **RegIRQ1**, **RegIRQ2** or **RegIRQ3** the contents of the addressed register are reset. Therefore one has to make a copy of the interrupt request register if there was more than one interrupt to treat. Each interrupt request flag may also be reset individually by writing 1 into it .

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 EM6627 goes into halt mode the **IntEn** bit is automatically set to 1, thus allowing to resume from halt mode with an interrupt.

9.1 Interrupt Control Registers

Table 9.1.9 Register RegIRQ1

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

*; Writing of 1 clears the corresponding bit.

Table 9.1.10 Register RegIRQ2

Bit	Name	Reset	R/W	Description
3	IRQHz1	0	R *	Prescaler interrupt request
2	IRQHz32/8	0	R *	Prescaler interrupt request
1	IRQBlink	0	R *	Prescaler interrupt request
0	IRQBz	0	R *	Melody timer interrupt request

*; Writing of 1 clears the corresponding bit.

Table 9.1.11 Register RegIRQ3

Bit	Name	Reset	R/W	Description
3	IRQSerial	0	R *	Serial interrupt request
2	IRQEEP	0	R *	EEP end of Write
1	IRQCount0	0	R *	Counter interrupt request
0	IRQCntComp	0	R *	Counter interrupt request

*; Writing of 1 clears the corresponding bit.

Table 9.1.12 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 9.1.13 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	MaskIRQBlink	0	R/W	Prescaler interrupt mask
0	MaskIRQBz	0	R/W	Melody timer interrupt mask

Interrupt is not stored if the mask bit is 0.

Table 9.1.14 Register RegIRQMask3

Bit	Name	Reset	R/W	Description
3	MaskIRQSerial	0	R/W	Serial interrupt mask
2	MaskIRQEEP	0	R/W	EEP end of Write
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

10. EEPROM (8 × 8 Bit)

The EEPROM addressing is indirect using 3 bits (8 addresses) defined in **RegEEPAdr** register.
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 and read or write. Refer to the examples below..

How to read data from EEPROM :

1st inst. : write EEPROM address in **RegEEPAddr** register.
2nd inst. : write reading and start operation in **RegEEPCntl.** (**EEPRdWr**='0', **EEPStart**='1')
Wait for Read time (NOP instructions or **EEPBusy*** polling)
3rd inst. : read EEPROM low data in **RegEEPDataL** register.
4th inst. : read EEPROM high data in **RegEEPDataH** register.
The two last instructions can be executed in the reverse order.

Wait time for read is 15 ck[16] periods long (typ 480us)This corresponds to 15 instruction cycles when the CPU is running at 32kHz and 225 Instructions if the CPU is running at 512kHz.

EEPVoltOk has no influence on the EEPROM reading.

How to write data in EEPROM :

A write access is always preceded by an erase of the selected memory byte.

Before writing to the EEPROM one must check that the supply voltage is high enough.
This check is performed by launching a VLD comparison selecting the the low level.
(write RegSvldCntl='010x , VLD start, low level, WD user defined).
Wait for the voltage check to finish before writing the EEPROM.

The result of this conversion can be read in register **RegVLDCntl** bit **VLDResult** and is stored in Register **RegEEPCntl** bit **EEPVoltOK**. To be able to modify the EEPROM contents, the bit **EEPVoltOK** must be at high level at the time when **EEPStart** is written high.
EEPVoltOk bit can be cleared by writing '1' to **EEPVoltClr**

1st inst. : write EEPROM address in **RegEEPAddr** register.
2nd inst. : write EEPROM low data in **RegEEPDataL** register.
3rd inst.: : write EEPROM high data in **RegEEPDataH** register.
4th inst. : write writing operation in **RegEEPCntl**. (**EEPRdWr**='1', **EEPStart**='1'), EEPBusy will go high
Wait for Write time (**EEPBusy*** polling or Halt mode with **IRQEEP** interrupt)
The first three instructions can be executed in any order.

Write takes a total of approx 15 ms (without the VLD check).

An **IRQEEP** is generated at the end of write and **EEPBusy** will go low.

Writing **EEPStart='1' in EEPCntl** register starts an EEPROM reading or writing operation according to the bit **EEPRdWr**.

Note: To change only one Nibble of EEDATA precede the EEP write by a EEP read in order to load the current value in EEPDataH,L registers. Then change the desired data-nibble and write back into EEP.

Note :

- *Any Reset or the sleep mode will immediately cancel the EEPROM write operation.*
- *An ongoing write may be corrupted.*
- ***Busy polling**: The busy flag will go high on next following system clock edge after start bit is written. Following dead times (start bit write until busy flag avilable) shall be observed.*
  CPU at 32kHz – 0 Instruction
  CPU at 64kHz – 1 instructions
  CPU at 128kHz – 3 instructions
  CPU at 256kHz – 7 instructions
  CPU at 512kHz – 15 Instructions

10.1 EEPROM registers

Table 10.1.1 EEPROM control register RegEEPCntl

Bit	Name	Reset	R/W	Description
3	-	-		
2	EEPVoltClr	0	W* (clear)	Clear EEPVolt bit
2	EEPVoltOk	0	R	Flag for sufficient EEP write power level
1	EEPStart	0	W	EEPROM Start read or Write
1	EEPBusy	0	R	EEPROM busy flag (write and read)
0	EEPRdWr	0	R/W	EEPROM operation read=0 / write=1

W*(Clear) A write access with EEPVoltClr=1 will clear the bit, write EEPVoltOk=0 has no action

Table 10.1.2 EEPROM address register RegEEPAdr

Bit	Name	Reset	R/W	Description
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 10.1.3 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 10.1.4 EEPROM data high register RegEEPDataH

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

10.2 EM6627 Trim values

The trim values for the RC Oscillator, VL1 and the VLD are written during the production test into the EEPROM. After system start-up the software shall copy this trim values in the respective trim registers to activate the trimming correction.

VL1 default value is in EEP at add 7 data[7:4], (MSB to LSB)
RC trim value at add 7 data[3:0], (MSB to LSB)
VLD level 0 trim value at add 6 data[1:0] (Up, down)
VLD level 1 trim value at add 6 data[5:4] (Up, down)

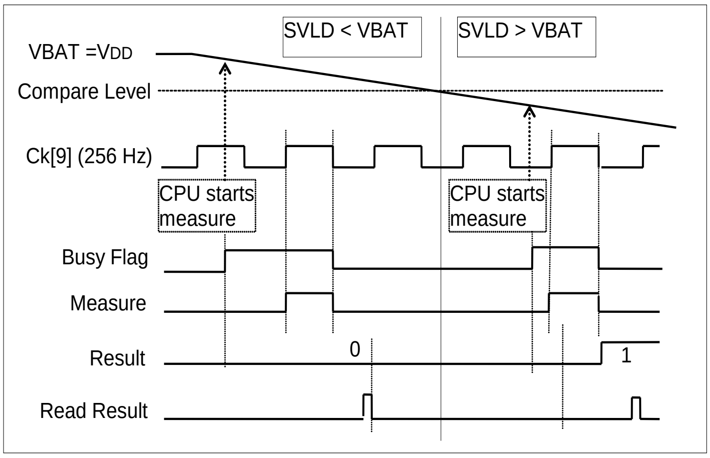
Default trim values in Register RegVL1trim and RegRCTrim are 0x00 (lowest RC freq and lowest VL1). It is recommended at software start to write this 2 registers to the middle value 0x07 and then copy - if necessary - the ideal trim value from the EEPROM into the trim registers. Same applies to the VLD level 0 and level 1 trim bits, if needed they shall be copied into the RegVldTrim register.

11. Supply Voltage Level Detector

The EM6627 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.
2 VLD levels are implemented. Level0 is typically 1.35V, Level1 2.4V. Selection with register bit VLDlevel (0=Level0, 1=Level1).

The 2 VLD levels are trimmed with an Up-bit and a Down-Bit as close as possible to the nominal value. The corresponding Up- and Down bits need to be copied from EEPROM into the SVLD trim register.
The user then has the choice to use the best nominal SVLD trim value or use the UP and down bits to introduce level thresholds. The typical hysteresis of an up and down bit is 50mV at Level0 and 200mV at Level1.

Figure 31. SVLD Timing Diagram



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 with bit **VldLevel** before the **VldStart** bit is set.
During the actual measurement (2 ms) the device will draw an additional 5 µA of IVDD current. After the end of the measure the result is available by inspection of the bit **VldResult**. If the result is read 0, then the power supply voltage was greater than the detection level value. If read 1, the power supply voltage was lower than the detection level value. During each read while **Busy=1** the **VldResult** is not guaranteed.

A VLD check with VLDLevel=0 selected will update the EEPVoltOk bit to allow EEPROM Write access.

The VLD trim words are in EEPROM adr6, Level 0 on low nibble, Level1 on high nibble.

11.1 SVLD Register

Table 11.1.1 Register RegVldCntl

Bit	Name	Reset	R/W	Description
3	VldResult	0	R*	Vld result flag
2	VldStart	0	W	Vld start
2	VldBusy	0	R	Vld busy flag
1	VLD level	0	R	'1' → Hi Level (Level1) '0' → lo level (Level0)
0	*NoLogicWD*	*0*	*R/W*	*No logic watchdog*

R*; Read value while VLDBusy=1 is not guaranteed.

Table 11.1.1 Register RegVldTrim

Bit	Name	POR	R/W	Description
3	-	-		Vld result flag
2	-	-		Vld start
1	VLDTrimUp	0	R/W	'1' → VLD level up
0	VLDTrimDown	0	R/W	'1' → VLD level down

Table 11.1.1 EEPROM control register RegEEPCntl

Bit	Name	Reset	R/W	Description
3	-	-	-	
2	EEPVoltClr	0	W* (clear)	Clear EEPVolt bit
2	EEPVoltOk	0	R	Flag for sufficient EEP write power level
1	*EEPStart*	*0*	*W*	*EEPROM Start read or Write*
1	*EEPBusy*	*0*	*R*	EEPROM busy flag (write and read)
0	*EEPRdWr*	*0*	*R/W*	*EEPROM operation read=0 / write=1*

W*(Clear) A write access with EEPVoltClr=1 will clear the bit, write EEPVoltOk=0 has no action

12. Strobe Output

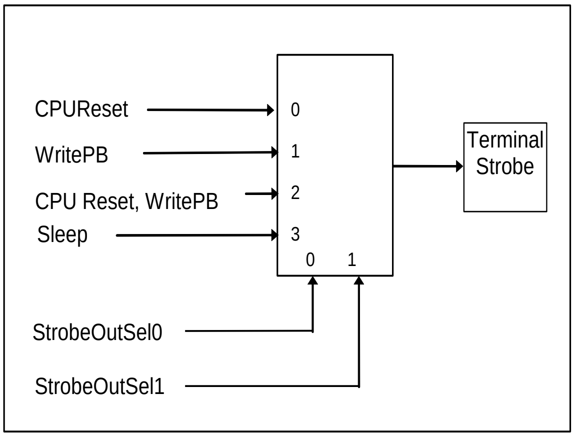
The Strobe output is used to indicate either the EM6627 CPU reset condition, a write operation on port B (WritePB) or the sleep mode. The selection is done in register **RegLcdCntl1**. Per default, the CPU reset condition is output on the Strobe terminal.
For a port B write operation the strobe signal 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 CPU 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 11.1.1. Strobe Output Selection

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

Figure 32 . Strobe Output



12.1 Strobe Register

Table 12.1.1 Register RegLCDCntl1

Bit	Name	power on value	R/W	Description
3	StrobeOutSel1	0	R/W	Strobe output select
2	StrobeOutSel0	0	R/W	Strobe output select
1	*LCDStaticDrive*	*0*	*R/W*	*LCD multiplier clock select*
0	*LcdPGO*	*0*	*R/W*	*LCD multiplier clock select*

13. RAM

The EM6627 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 33. 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 33. 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. Unused LCD register latches can also be used as RAM memory.

14. Program Memory

14.1 Rom Version

The ROM size is 4096 Instructions of 16bit each. It holds the user instructions.

15. LCD Driver

The EM6627 has a built-in Liquid Crystal Display (LCD) driver. A maximum of 80 Segments can be displayed using the 20 Segment driver outputs (SEG[20:1) in 4:1 multiplex[Com4:1] ,60 Segments in the case of 3:1 multiplex[Com3:1], and 20 segments in case of Static drive[Com4:1 of equal value].
The LCD driver has its own voltage regulator (1.05 Volt) and voltage multiplier to generate the driver bias voltages VL1, VL2 and VL3 (also called VLCD).

The LCD clock frequency is 256Hz. The resulting frame frequency is 256/8 Hz if 4:1 multiplex, or 256/6 if 3:1 multiplex or 128/2 in static drive. In static drive all the 4 COM outputs drive the same backplane signal and can be connected together externally if high output drives are needed.

The LCD voltage bias is based on a reference of 1.05V (internal reference). This reference voltage can be user adjusted by register RegVL1Trim. These settings have no influence when using external bias voltages.

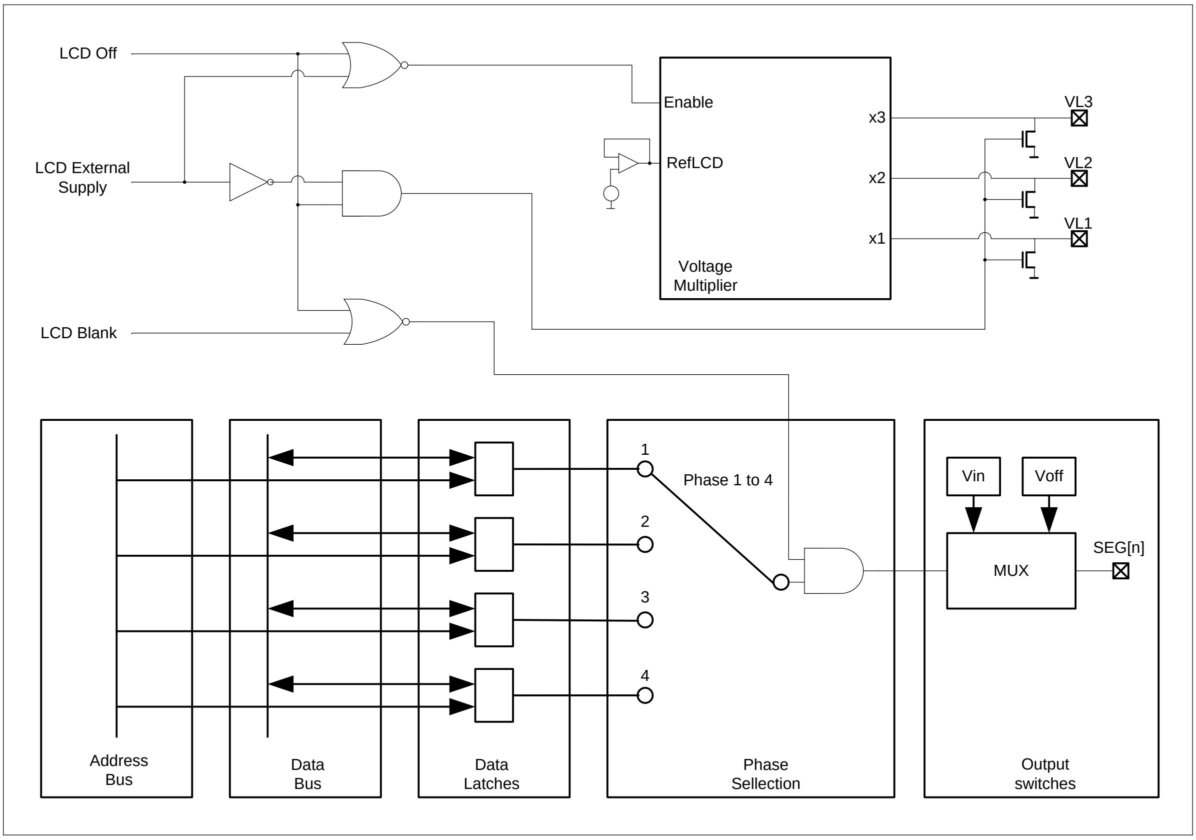
Example1: LCD 3 times mux with integrated voltage multiplier
1st: RegLcdCntl2=1000 (Blank, LCD on, 3mux, internal supply:
→ multiplier starts up, Seg, Com active)
2nd: Write display data into LCD data registers
3rd: RegLcdCntl2=0000 (remove LCD blank after the multiplier voltages are stabilized, Display on)

Example2: LCD 4 times mux with external supply
1st: RegLcdCntl2=1011 (Blank, LCD on, 4mux, external supply:
→ Seg, Com active)
2nd: Write display data into LCD data registers
3rd: RegLcdCntl2=0011 (remove LCD blank, Display on)

Example3: LCD static drive with external supply
1st: RegLcdCntl1=xx10 (static drive setup but lcd off)
2nd: initialize the LCD data registers
3rd: RegLcdCntl2=00x1 (Display on in static drive mode)

Example4: LCD static drive with internal supply
1st: RegLcdCntl1=xx10 (static drive setup but lcd off)
2nd: RegLcdCntl2=1010 (LCD on , multiplier starts up, display blank)
3rd: initialize the LCD data registers
4th: RegLcdCntl2=00x0 (Display on in static drive mode, internal supply)

Figure 34. LCD Architecture (3 or 4 times mux)



LCD Control

The LCD driver has two control registers **RegLCDCntl1**, **RegLCDCntl2** to optimize for display contrast, power consumption, operation mode and bias voltage source.

LCDExtSupply: Choosing external supply (**LCDExtSupply** ='1') disables the internal LCD voltage regulator and voltage multiplier, it also puts the bias voltage terminals VL1, VL2 and VL3 into high impedance state. External bias levels can now be connected to VL1, VL2 and VL3 terminals. (Resistor divider chain or others).

Another way to adapt the VL1, VL2 and VL3 levels to specific user needs is to overdrive the VL1 output **(LCDExtSupply** =0) with the desired value. The internal multiplier will multiply this new VL1 level to generate the corresponding levels VL2 and VL3. The bit **LCDExtSupply** is only reset by initial POR.

LCD4Mux: With this switch one selects either 3:1 or 4:1 (default) times multiplexing of the 20 Segment driver outputs. In the case of 3:1 multiplexing the COM[4] is off.
If LCD4MUX=1 the lcd data register bits 0 will be output in phase1, bits 1 in phase 2, bits 2 in phase 3 and bits 4 in phase 4.
If LCD4MUX=3 the lcd data register bits 0 will be output in phase1, bits 1 in phase 2 and bits 2 in phase 3. Bit 4 can be used for other purposes.
Static drive has priority over LCD4Mux setting.

LCDOff: Disables the LCD. The voltage multiplier and regulator are switched off (0 current).The Segment latch information is maintained. The VL1,VL2 and VL3 outputs are pulled to VSS, except if LCDExtSupply is '1'.

LCDBlank: All Segment outputs are turned off. The voltage multiplier and regulator remain switched on. **LCDBlank** can be used with the 1Hz and Blink interrupt to let the whole display blink (software controlled).

LCDStaticDrive: Will set the 20 segment outputs in static drive mode. The phase is locked on phase 1. The Com[1] value is copied to COM[2], Com[3] Com[4]. (Can be externally put together to increase the drive strength).
In static drive the lcd data register data bits 0 will be output. The other lcd data register bits can be freely used.
Static drive has priority over Lcd4Mux settings.

15.1 LCD Display ON/Off Voltages (RMS values)

	ON voltage [VL1]	ON voltage [VL3]	OFF voltage [VL1]	Off voltage [VL3]
LCD – 4Mux	1.5 * VL1	0.5 * VL3	VL1	1/3 * VL3
LCD – 3Mux	5/3 * VL1	5/9 * VL3	VL1	1/3 * VL3
LCD – static drive	3 * VL1	VL3	0	0

15.2 LCD Addressing

The LCD driver addressing is indirect using the **RegIndexAdr** value as an offset to the directly addressed base **LCD_1**, **LCD_2** registers. All LCD Segment registers are R/W. At address LCD_1 all 16 Index locations are usable and are R/W.

At address LCD_2 only the first 4 Index locations are usable. The Index locations hex(4 to F) are non implemented and can not be used as RAM.

Figure 35. LCD Address Mapping

20 x 4 Indexed Addressable LCD Latches		
	-	-
	RegIndexAdr[3]	4 bit R/W
	RegIndexAdr[2]	4 bit R/W
	RegIndexAdr[1]	4 bit R/W
LCD_2	RegIndexAdr[0]	4 bit R/W
	RegIndexAdr[F]	4 bit R/W
	RegIndexAdr[E]	4 bit R/W
	...	...
	RegIndexAdr[1]	4 bit R/W
LCD_1	RegIndexAdr[0]	4 bit R/W

15.3 Segment Allocation

Each Segment (SEG[20:1]) terminal outputs the time multiplexed information from its 4 Segment data latches. Information stored in latch 1 is output during phase1, latch 2 during phase 2, latch 3 during phase 3 and latch 4 during phase 4. In the case of 3 to 1 multiplexing the phase 4 and the latch 4 are not used. This phase information on the segment outputs together with the common outputs (COM[4:1]) - also called back-planes - defines if a given LCD segment is light or not. COM[1] is on during phase 1 and off during phase 2,3,4 , COM[2] is on during phase 2 and off during phase 1,3,4 , etc.
In case of static drive only the phase1 is used and will output the information stored in latch 1.

Table 15.3.1 LCD Configuration (4 mux)

Segment outputs	COM[1] = phase1	COM[2] = phase2	COM[3] = phase3	COM[4] = phase4
SEG[1]	DB[0], LCDAdr[0]	DB[1], LCDAdr[0]	DB[2], LCDAdr[0]	DB[3], LCDAdr[0]
SEG[2]	DB[0], LCDAdr[1]	DB[1], LCDAdr[1]	DB[2], LCDAdr[1]	DB[3], LCDAdr[1]
SEG[3]	DB[0], LCDAdr[2]	DB[1], LCDAdr[2]	DB[2], LCDAdr[2]	DB[3], LCDAdr[2]
...	...	...	...	...
SEG[18]	DB[0], LCDAdr[17]	DB[1], LCDAdr[17]	DB[2], LCDAdr[17]	DB[3], LCDAdr[17]
SEG[19]	DB[0], LCDAdr[18]	DB[1], LCDAdr[18]	DB[2], LCDAdr[18]	DB[3], LCDAdr[18]
SEG[20]	DB[0], LCDAdr[19]	DB[1], LCDAdr[19]	DB[2], LCDAdr[19]	DB[3], LCDAdr[19]

Table 15.3.1 LCD Configuration (3 mux)

Segment outputs	COM[1] = phase1	COM[2] = phase2	COM[3] = phase3	Free available data
SEG[1]	DB[0], LCDAdr[0]	DB[1], LCDAdr[0]	DB[2], LCDAdr[0]	DB[3], LCDAdr[0]
SEG[2]	DB[0], LCDAdr[1]	DB[1], LCDAdr[1]	DB[2], LCDAdr[1]	DB[3], LCDAdr[1]
SEG[3]	DB[0], LCDAdr[2]	DB[1], LCDAdr[2]	DB[2], LCDAdr[2]	DB[3], LCDAdr[2]
...	...	...	...	...
SEG[18]	DB[0], LCDAdr[17]	DB[1], LCDAdr[17]	DB[2], LCDAdr[17]	DB[3], LCDAdr[17]
SEG[19]	DB[0], LCDAdr[18]	DB[1], LCDAdr[18]	DB[2], LCDAdr[18]	DB[3], LCDAdr[18]
SEG[20]	DB[0], LCDAdr[19]	DB[1], LCDAdr[19]	DB[2], LCDAdr[19]	DB[3], LCDAdr[19]

only data bit 0,1,3 of each LCDAdr is output, the r data bit 3 bits can be used a free data space.

Table 15.3.1 LCD Configuration (Static drive)

Segment outputs	COM[1], Com[2], Com[3], Com[4] = phase1	Free available data	Free available data	Free available data
SEG[1]	DB[0], LCDAdr[0]	DB[1], LCDAdr[0]	DB[2], LCDAdr[0]	DB[3], LCDAdr[0]
SEG[2]	DB[0], LCDAdr[1]	DB[1], LCDAdr[1]	DB[2], LCDAdr[1]	DB[3], LCDAdr[1]
SEG[3]	DB[0], LCDAdr[2]	DB[1], LCDAdr[2]	DB[2], LCDAdr[2]	DB[3], LCDAdr[2]
...	...	...	...	...
SEG[18]	DB[0], LCDAdr[17]	DB[1], LCDAdr[17]	DB[2], LCDAdr[17]	DB[3], LCDAdr[17]
SEG[19]	DB[0], LCDAdr[18]	DB[1], LCDAdr[18]	DB[2], LCDAdr[18]	DB[3], LCDAdr[18]
SEG[20]	DB[0], LCDAdr[19]	DB[1], LCDAdr[19]	DB[2], LCDAdr[19]	DB[3], LCDAdr[19]

only data bit 0 of each LCDAdr is output, the other data bits can be used a free data space.

15.4 LCD Registers

Table 15.4.1 Register RegLcdCntl1

Bit	Name	Reset	R/W	Description
3	*StrobeOutSel1*	*POR to '0'*	*R/W*	*Strobe output select*
2	*StrobeOutSel0*	*POR to '0'*	*R/W*	*Strobe output select*
1	LCDStaticDrive	*POR to '0'*	R/W	LCD Static drive setting
0	-	*POR to '0'*	-	Reserved for EM general purpose

StrobeOutSel1,0 is reset by initial power on only.

Table 15.4.2 Register LcdCntl2

Bit	Name	Reset	R/W	Description
3	LCDBlank	1	R/W	LCD Segment outputs off '0' output in GPO mode
2	LCDOff	1	R/W	LCD off (multiplier off) Seg and Com terminals all tristate
1	LCD4Mux	1	R/W	4 : 1 multiplexed
0	LCDExtSupply	POR to '1'	R/W	External supply for VL1, VL2 and VL3

LCDExtSupply is reset to '1' by POR only.

Table 15.4.3 Register VL1Trim

Bit	Name	POR	R/W	Description
3	VL1trim[3]	0	R/W	VL1 trim bit, MSB
2	VL1trim[2]	0	R/W	VL1 trim bit
1	VL1Trim[1]	0	R/W	VL1 trim bit
0	VL1Trim[0]	0	R/W	VL1 trim bit, LSB

Figure 36. Triming range of VL1 voltage level and VL1, 2, 3 temperature dependencies

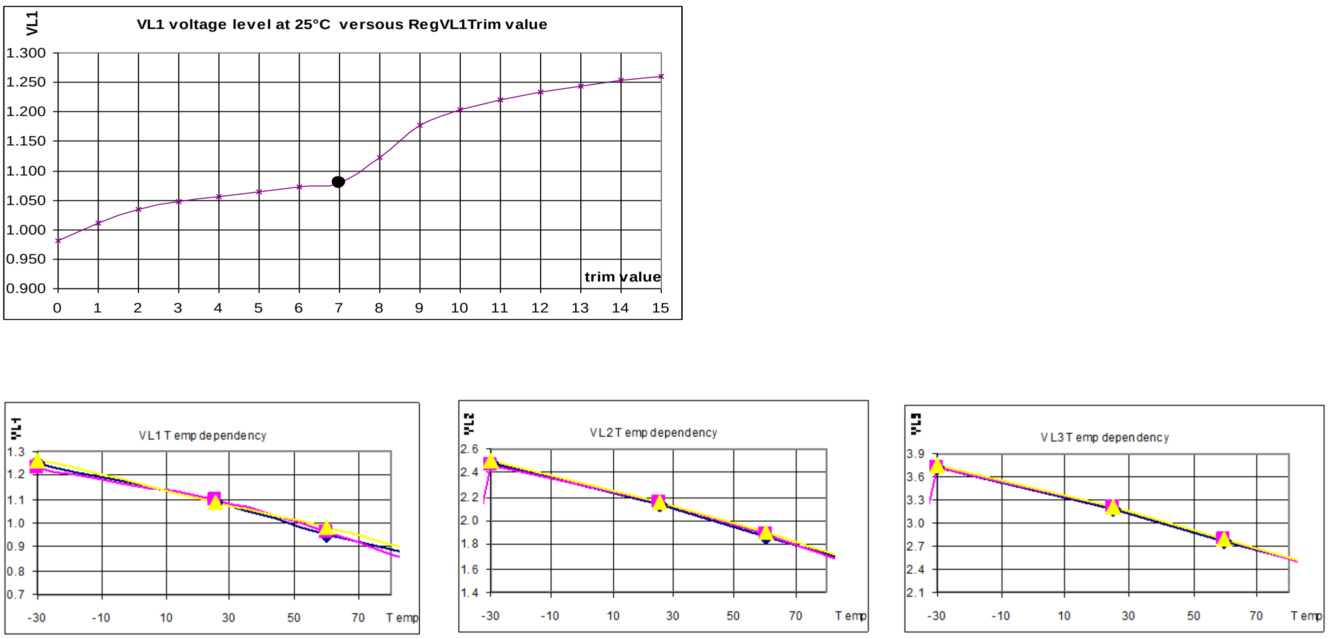


Figure 37 LCD Multiplexing Waveform – 4Mux

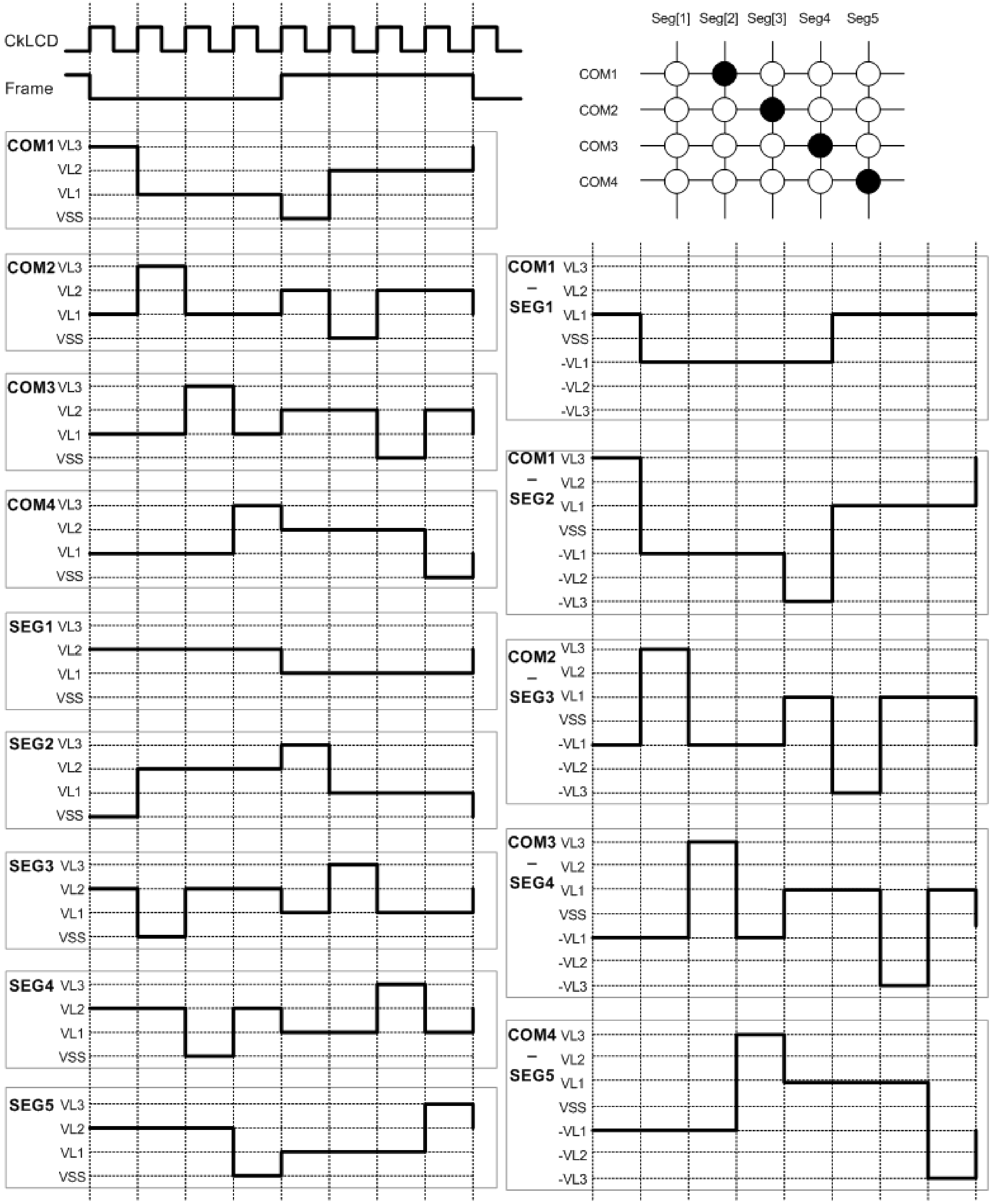


Figure 38 LCD Multiplexing Waveform – 3 Mux

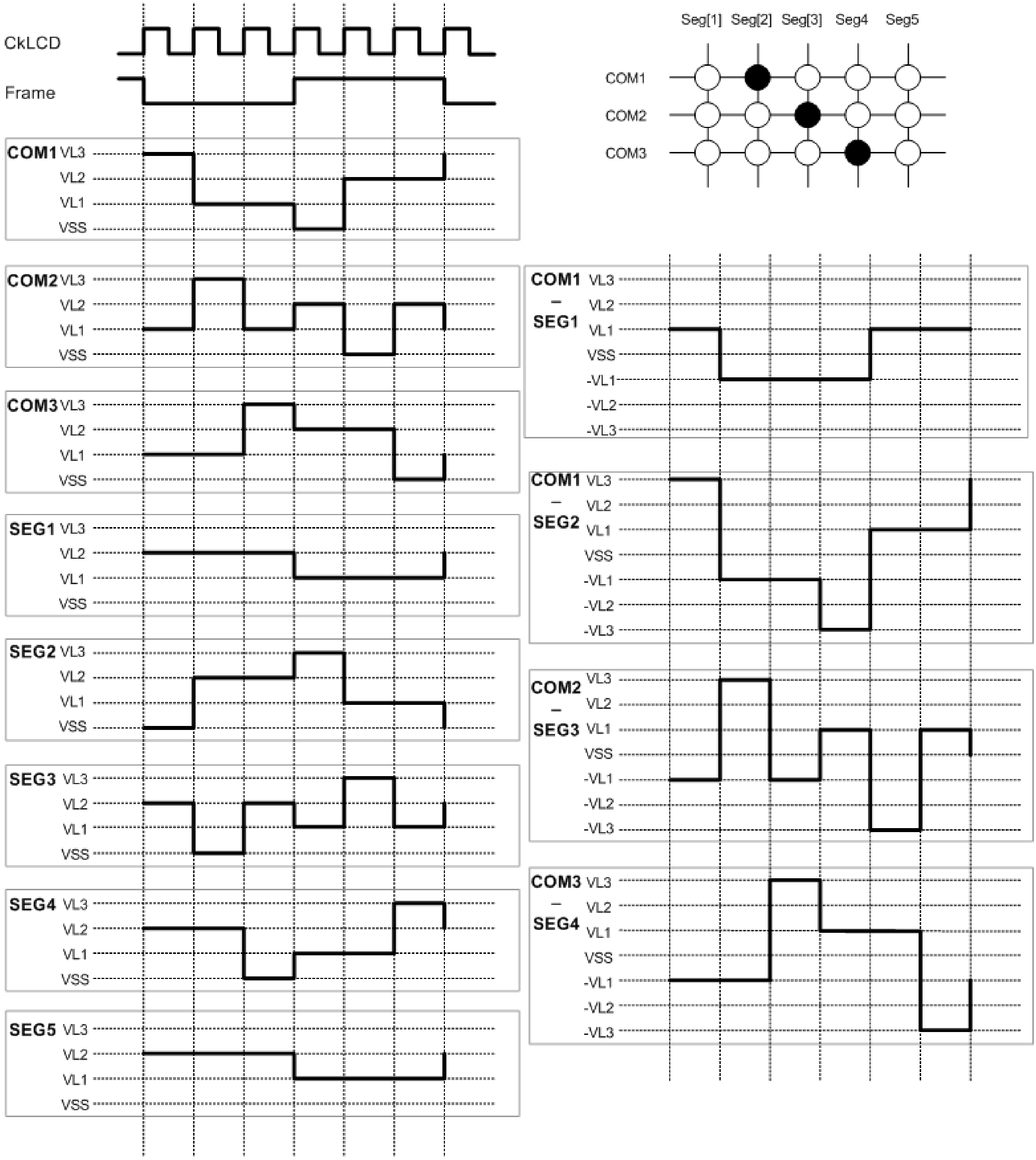
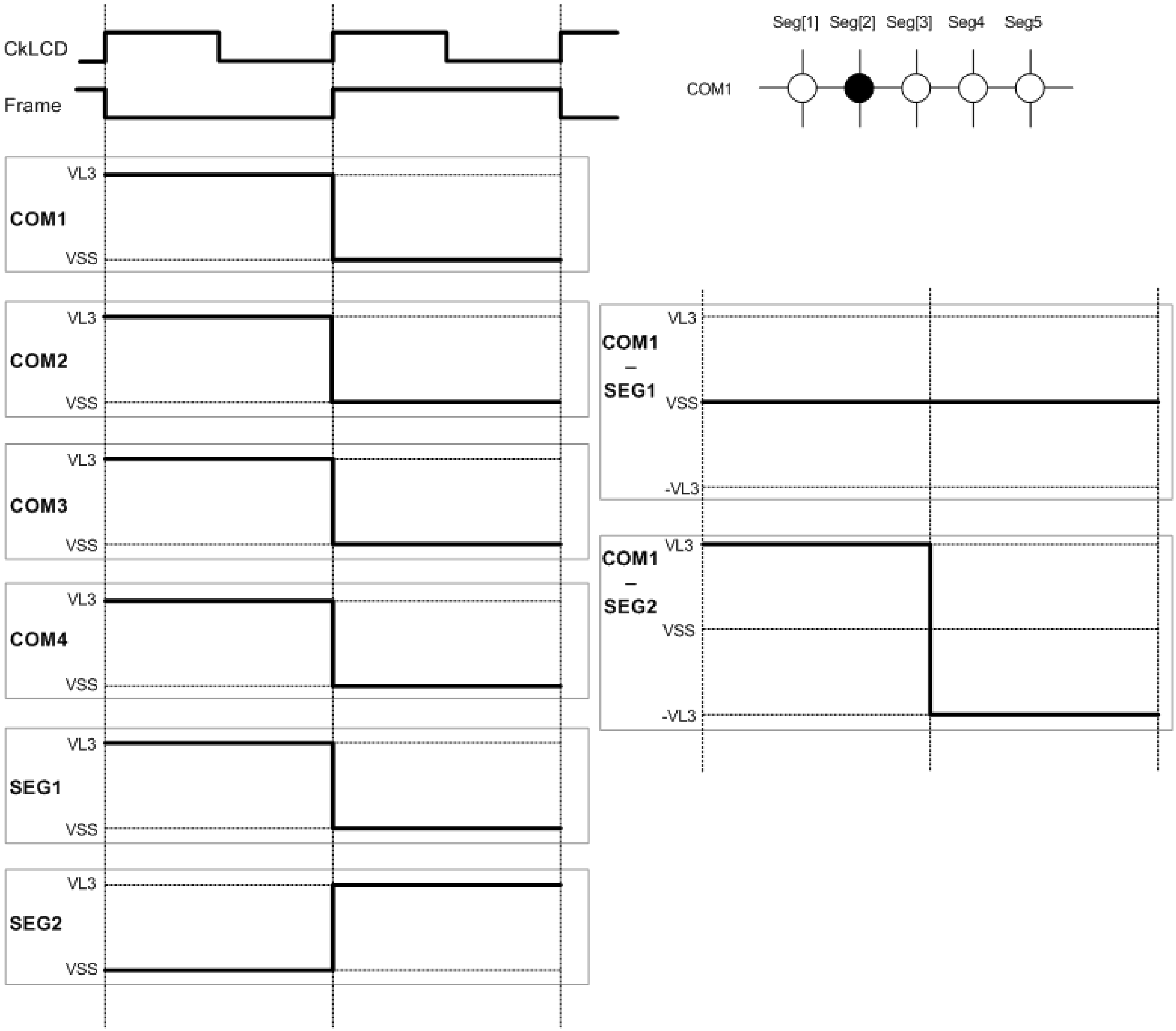


Figure 39 LCD Multiplexing Waveform, static drive



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	RamData[3:0]		
…						
Ram1_63	3F	63	xxxx	RamData[3:0]		
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
LCD_1	44	68	xxxx	0: Data0 1: Data1 2: Data2 3: Data3		16 nibbles addressable over index register on add 'H70
LCD_2	45	69	xxxx	0: Data0 1: Data1 2: Data2 3: Data3		The 4 lower nibbles are addressable over the index register on add 'H70. The 12 higher Nibbles are not used and not implemented
	46	70	xxxx			Reserved, not implemented
	47	71				Reserved, not implemented
	48	72				Reserved, not implemented
	49	73				Reserved, not implemented
	4A	74				Reserved, not implemented
RegTmodeEE1	4B	75	-110	*0: EEP_MARGV* *1:EEP_MARGL* *2:EEP_MARGH* *3:-*		*EM Test Register* *In reset state while test=0*
RegTmodeEE2	*4C*	*76*	*01-0*	*0: ChTmDis* *1:-* *2:EEP_WSEL1R* *3:EEP_WSEL2R*		*EM Test Register* *In reset state while test=0*
RegVldTrim	4D	77	--pp	VldTrim[1:0]		Vld correction
RegRCTrim	4E	78	pppp	RCTrim[3:0]		RegRCTrim
RegVL1Trim	4F	79	pppp	VL1Trim[3:0]		VL1 Trimming
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

P = defined by POR (power on reset) to ‘1’
p = defined by POR (power on reset) to ‘0’

Register Name	Add Hex	Add Dec.	Reset Value b'3210	Read Bits	Write Bits	Remarks
				Read / Write Bits		
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
RegSCntl1	53	83	0000	0: MSBnLSB 1: POSnNeg 2: MS0 3: MS1		Serial interface control 1
RegSCntl2	54	84	0000	0: OM[0] 1: OM[1] 2: Status 3: Start		Serial interface control 2
RegSDataL	55	85	0000	0: SerDataL[0] 1: SerDataL[1] 2: SerDataL[2] 3: SerDataL[3]		Serial interface low data nibble
RegSDataH	56	86	0000	0: SerDataH[0] 1: SerDataH[1] 2: SerDataH[2] 3: SerDataH[3]		Serial interface high data nibble
RegSPData	57	87	0000	0: PSP[0] 1: PSP[1] 2: PSP[2] 3: PSP[3]	0: SerPData[0] 1: SerPData[1] 2: SerPData[2] 3: SerPData[3]	Serial interface parallel data out
RegMelFSel	58	88	0000	0: MelFSel[0] 1: MelFSel[1] 2: MelFSel[2] 3: BzOutEn		Melody frequency select and output enable control
RegMelTim	59	89	0000	0:FTimSel0 1:FTimSel1 2:Auto 3:FlBuzzer	0:FTimSel0 1:FTimSel1 2:Auto 3:SwBuzzer	Melody timer control
RegMelPeri	5A	90	0000	0: - 1: - 2: - 3: -	0: Per[0] 1: Per[1] 2: Per[2] 3: Per[3]	Melody timer period
RegCCntl1	5B	91	0000	0: CountFSel0 1: CountFSel1 2: CountFSel2 3: UP/Down		10-bit counter control 1; frequency and up/down
RegCCntl2	5C	92	0000	0: '0' 1: EnComp 2: EvCount 3: Start	0 : Load 1: EnComp 2: EvCount 3: Start	10-bit counter control 2; comparison, event counter and start
RegCDataL	5D	93	0000 (up) 1111(down)	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 compare data low nibble
RegCDataM	5E	94	0000 (up) 1111(down)	0: ShCount[4] 1: ShCount[5] 2: ShCount[6] 3: ShCount[7]	0: CReg[4] 1: CReg[5] 2: CReg[6] 3: CReg[7]	10-bit counter compare data middle nibble
RegCDataH	5F	95	0000 (up) 0011(down)	0: ShCount[8] 1: ShCount[9] 2: BitSel[0] 3: BitSel[1]	0: CReg[8] 1: CReg[9] 2: BitSel[0] 3: BitSel[1]	10 bit counter compare data high bits , counter length
RegRCSel	60	96	pppp	0: RCSel[0] 1: RCSel[1] 2: RCSel[2] 3: RCSel[3]		RC divider settings

P = defined by POR (power on reset) to '1'
p = defined by POR (power on reset) to '0'

Register Name	Add Hex	Add Dec.	Reset Value b'3210	Read Bits	Write Bits	Remarks
				Read / Write Bits		
RegEEPCntl	61	97	0000	0: EEPRdWr 1: EEPBusy 2: EEPVoltOk 3: -	0: EEPRdWr 1: EEPStart 2: EEPVoltClr 3:-	EEPROM control register
RegEEPADr	62	98	0000	0: EEPAdr[0] 1: EEPAdr [1] 2: EEPAdr [2] 3: -		EEPROM address register
RegEEPDataL	63	99	0000	EEPDataL[3:0]		EEPROM R/W low nibble
RegEEPDataH	64	100	0000	EEPDataH[3:0]		EEPROM R/W high nibble
RegIRQMask1	65	101	0000	0: MaskIRQPA[0] 1: MaskIRQPA[1] 2: MaskIRQPA[2] 3: MaskIRQPA[3]		Port A interrupt mask; masking active 0
RegIRQMask2	66	102	0000	0: MaskIRQBz 1: MaskIRQBlink 2: MaskIRQHz32/8 3: MaskIRQHz1		Buzzer and prescaler interrupt mask; masking active low
RegIRQMask3	67	103	0000	0: MaskIRQCntComp 1: MaskIRQCount0 2: MaskIRQEEP 3: MaskIRQSerial		10-bit counter, EEPROM, serial 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 interrupt Write: Reset IRQ if data bit = 1.
REgIRQ2	69	105	0000	0: IRQBz 1: IRQBlink 2: IRQHz32/8 3: IRQHz1	0:RIRQBz 1:RIRQBlink 2:RIRQHz32/8 3:RIRQHz1	Read: buzzer and prescaler IRQ ; Write: Reset IRQ id data bit = 1
RegIRQ3	6A	106	0000	0:IRQCntComp 1: IRQCount0 2: IRQEEP 3: IRQSerial	0:RIRQCntComp 1:RIRQCount0 2:RIRQEEP 3:RIRQSerial	Read: 10-bit counter, EEPROM, serial interrupt Write: Reset IRQ if data bit =1.
RegSysCntl1	6B	107	0000	0: PWMOn 1: SelIntFull 2: '0' 3: IntEn	0: PWMOn 1: SelIntFull 2: Sleep 3: IntEn	System Control 1; Counter PWM setups, Sleep and general Interrupt
RegSysCntl2	6C	108	0p00	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
RegPresc	6D	109	0000	0: DebSel 1: PrIntSel 2: '0' 3: HiDebCkSel	0: DebSel 1: PrIntSel 2: ResPresc 3: HiDebCkSel	Prescaler control; debouncer and prescaler interrupt select
IXLow	6E	110	xxxx	0: IXLow[0] 1: IXLow[1] 2: IXLow[2] 3: IXLow[3]		Internal µP index register low nibble; for µP indexed addressing

P = defined by POR (power on reset) to '1'
p = defined by POR (power on reset) to '0'

<table>
<tr><th>Register Name</th><th>Add Hex</th><th>Add Dec.</th><th>Reset Value</th><th>Read Bits</th><th>Write Bits</th><th rowspan="2">Remarks</th></tr>
<tr><th colspan="3"></th><th>b'3210</th><th colspan="2">Read / Write Bits</th></tr>
<tr><td>IXHigh</td><td>6F</td><td>111</td><td>xxxx</td><td>0: IXHigh[4]
1: IXHigh[5]
2: IXHigh[6]
3: '0'</td><td>0: IXHigh[4]
1: IXHigh[5]
2: IXHigh[6]
3: --</td><td>Internal µP index register high nibble; for µP indexed addressing</td></tr>
<tr><td>RegIndexAdr</td><td>70</td><td>112</td><td>0000</td><td colspan="2">0: IndexAdr[0]
1: IndexAdr[1]
2: IndexAdr[2]
3: IndexAdr[3]</td><td>Indexed addressing register for 4x16 nibble RAM2 and 1x16 + 4 nibble LCD</td></tr>
<tr><td>RegLCDCntl1</td><td>71</td><td>113</td><td>pppp</td><td colspan="2">0: --
1: LCDStaticDrive
2: StrobeOutSel0
3: StrobeOutSel1</td><td>LCD control 0; multiplier clock and strobe output select</td></tr>
<tr><td>RegLCDCntl2</td><td>72</td><td>114</td><td>111P</td><td colspan="2">0: LCDExtSupply
1: Lcd4xMux
2: LCDOff
3: LCDBlank</td><td>LCD control 1; main selects</td></tr>
<tr><td>RegVldCntl</td><td>73</td><td>115</td><td>0000</td><td>0: NoLogicWD
1: Vldlevel
2: VldBusy
3: VldResult</td><td>0: NoLogicWD
1: Vldlevel
2: VldStart
3: --</td><td>Voltage level detector control</td></tr>
</table>

P = defined by POR (power on reset) to '1'
p = defined by POR (power on reset) to '0'

17. IO Configuration Register Mapping

The values of the hardware configuration 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, reset from pin RESET, etc. do not change the configuration register value.

Register Name	Add Hex	Add Dec.	Reset Value b'3210	Read Bits	Write Bits	Remarks
				Read / Write Bits		
CRegPuPA	74	116	0000	0: PuPA[0] 1: PuPA[1] 2: PuPA[2] 3: PuPA[3]		PortA pullup settings
CRegDebIntPA	75	117	0000	0: NoDebIntPA[0] 1: NoDebIntPA[1] 2: NoDebIntPA[2] 3: NoDebIntPA[3]		Debouncer on port A for interrupt gen. Default: debouncer on
CRegIntEdgPA	76	118	0000	0: IntEdgPA[0] 1: IntEdgPA[1] 2: IntEdgPA[2] 3: IntEdgPA[3]		Interrupt edge select on port A. Default: pos. edge
CRegNoPdPA	77	119	0000	0: NoPdPA[0] 1: NoPdPA[1] 2: NoPdPA[2] 3: NoPdPA[3]		Pull-down selection on port A Default: pull-down
CRegNoPdPB	78	120	0000	0: NoPdPB[0] 1: NoPdPB[1] 2: NoPdPB[2] 3: NoPdPB[3]		Pull-down selection on port B Default: pull-down
CRegNchOpDPB	79	121	0000	0: NchOpDPB[0] 1: NchOpDPB[1] 2: NchOpDPB[2] 3: NchOpDPB[3]		Nch. open drain output on port B Default: CMOS output
CRegNchOpDPS	7A	122	0000	0: NchOpDPS[0] 1: NchOpDPS[1] 2: NchOpDPS[2] 3: NchOpDPS[3]		Nch. open drain output on port serial Default: CMOS output
CRegFSelPB	7B	123	0000	0: PB1HzOut 1: PB1kHzOut 2: PB32kHzOut 3: InpResSleep		Frequency output on port B, reset from sleep mode with port A
CRegInpRSel1	7C	124	0000	0: InpRes1PA[0] 1: InpRes1PA[1] 2: InpRes1PA[2] 3: InpRes1PA[3]		Reset through port A inputs selection. Refer to reset part
CRegInpRSel2	7D	125	0000	0: InpRes2PA[0] 1: InpRes2PA[1] 2: InpRes2PA[2] 3: InpRes2PA[3]		Reset through port A inputs selection. Refer to reset part
CRegNoPdPS	7E	126	0000	0: NoPdPS[0] 1: NoPdPS[1] 2: NoPdPS[2] 3: NoPdPS[3]		No Pull-down on port SP Default: pull-down
RegTestEM	*7F*	*127*	----	----	----	*for EM test only;*

18. Active Supply Current Test

For this purpose, five instructions at the end of the ROM will be added at EM Marin
As such the user programming area is from address 0 to address 4090.

```
TESTLOOP : STI     00H,    0AH            ;TEST LOOP
           LDR     1BH
           NORX
           JPZ     TESTLOOP
           JMP     0
```

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

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

Note: This test loop is intended for EM internal test only.

18.1 Empty ROM space

- Free space after last user instruction is filled with Instruction JMP 00H (0x0000)
- *Empty space within the program area is filled with the instruction NOP (0xFOFF).*

19. Mask Options

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

19.1.1 Voltage Regulator Option

Option Name		Default Value	User Value
MVreg	Voltage Regulator	**YES**	

Possible values: Yes, No

By default **MVreg(Yes)** the internal voltage regulator supplies the core logic the RAM and the ROM.
Setting **MVreg(No)** the regulator is cut and Vbat is supplying the core logic the RAM and the ROM.

19.1.2 RC Oscillator frequency selection

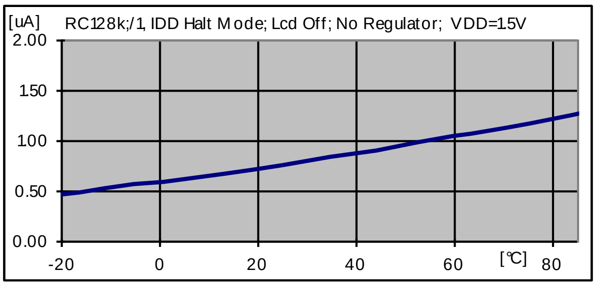
Option Name		Default Value	User Value
MRCFreq	RC Osc frequency	**512kHz**	

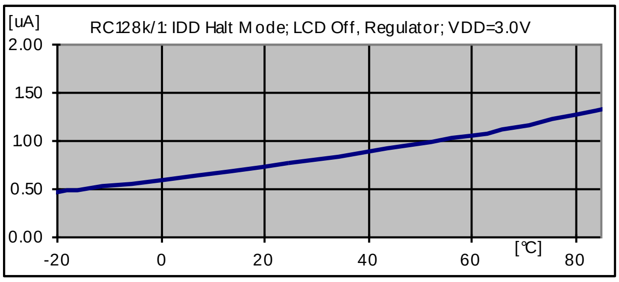
Possible values: 512kHz, 128kHz

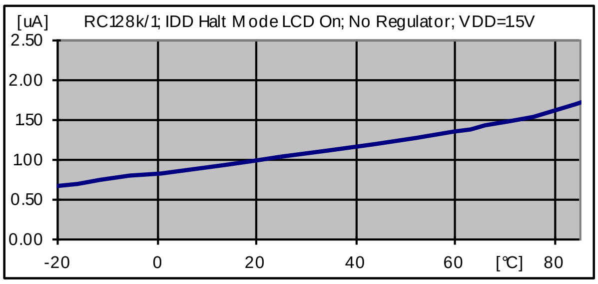
By default **MRCFreq(512kHz)** the RC Oscillator runs on a base frequency of 512kHz.
Setting **MRCFReq(128kHz)** the RC Oscillator runs on a base frequency of 128kHz.

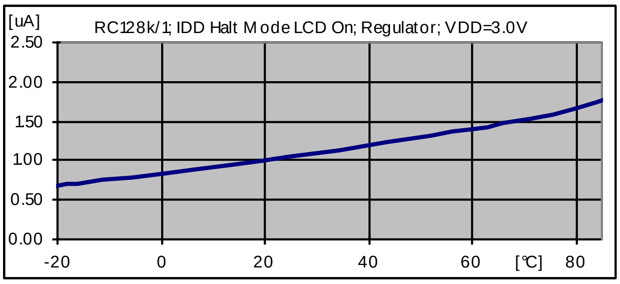
20. Temp. and Voltage Behaviors

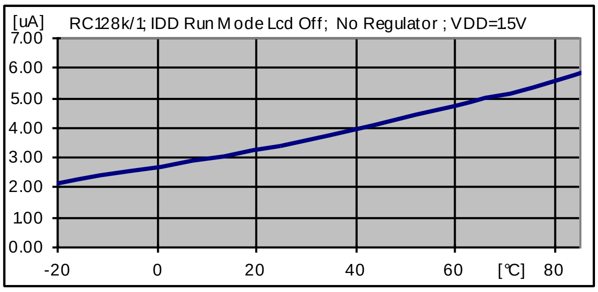
20.1 IDD Current (RC 128 kHz, CPUClk 128kHz/1)

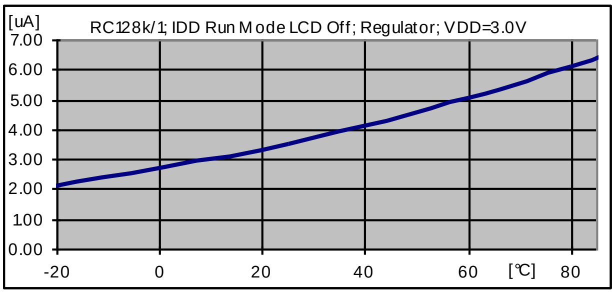


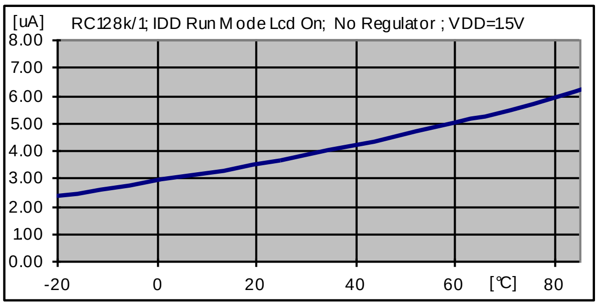


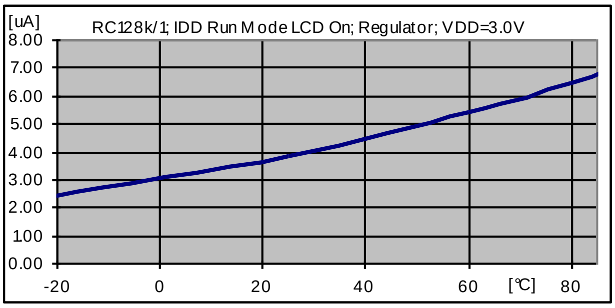


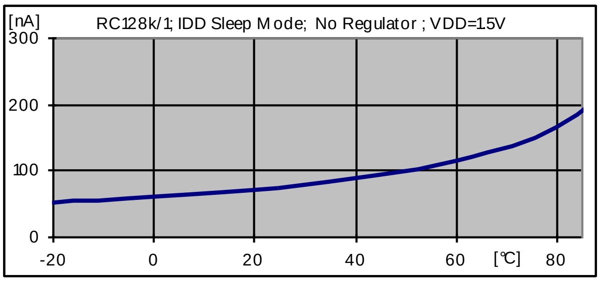


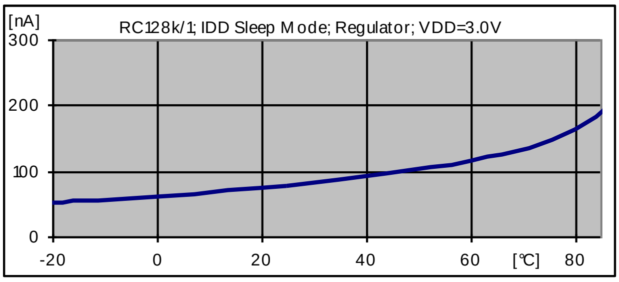




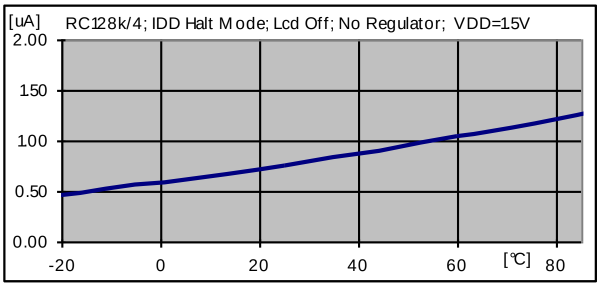


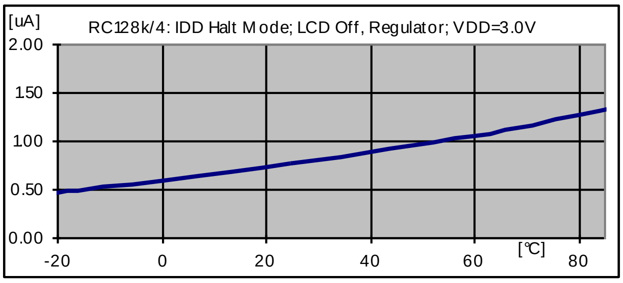


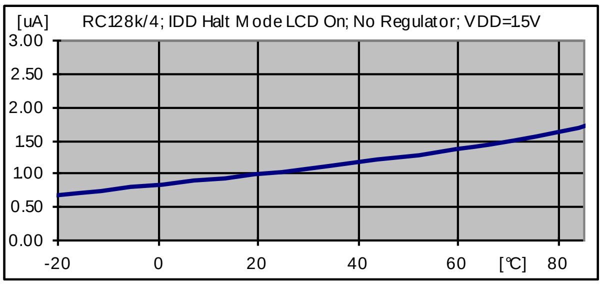


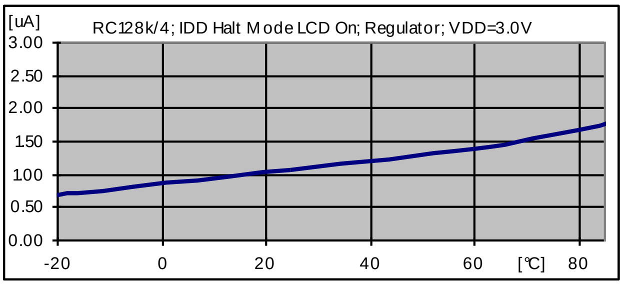


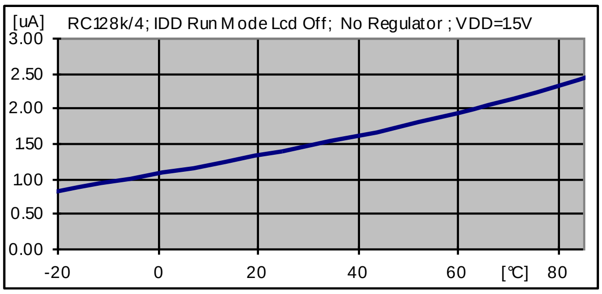
20.2 IDD Current (RC 128 kHz, CPUClk 128kHz/4)

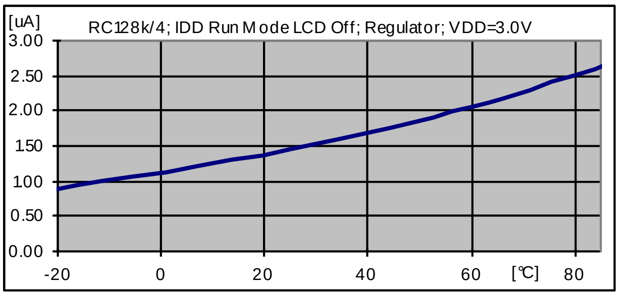


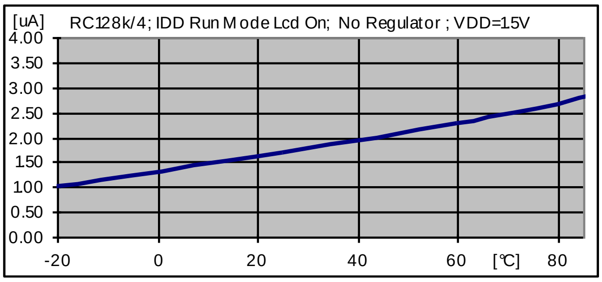


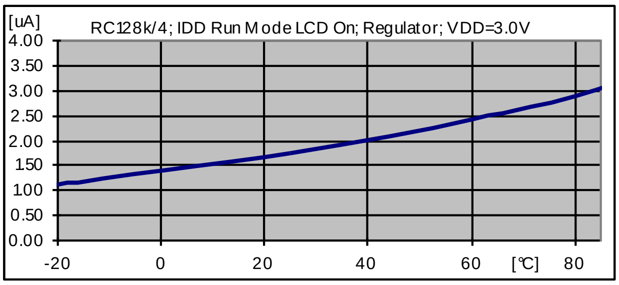


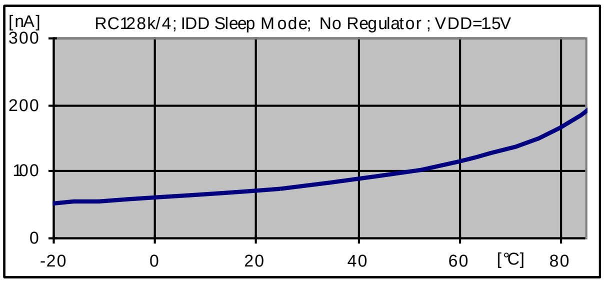


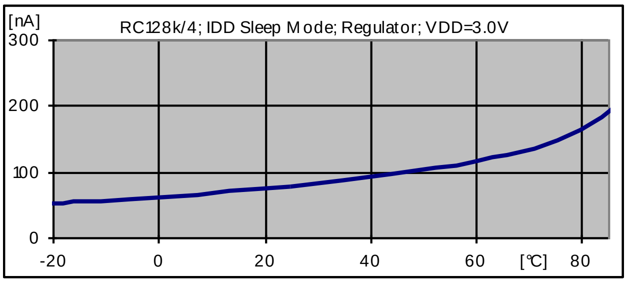




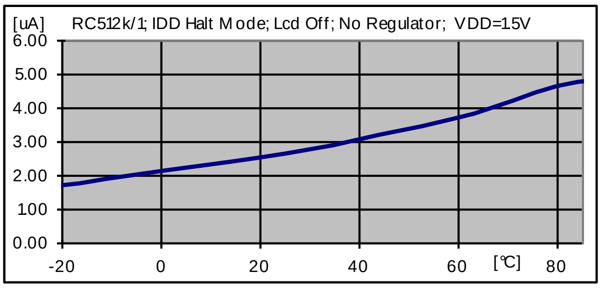


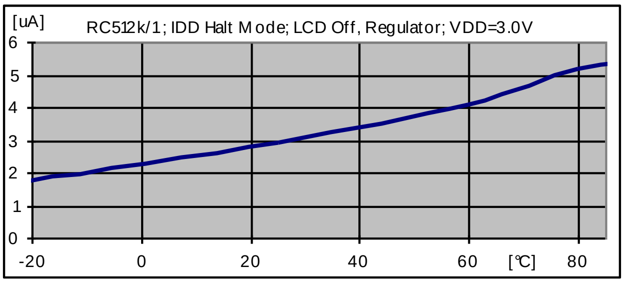


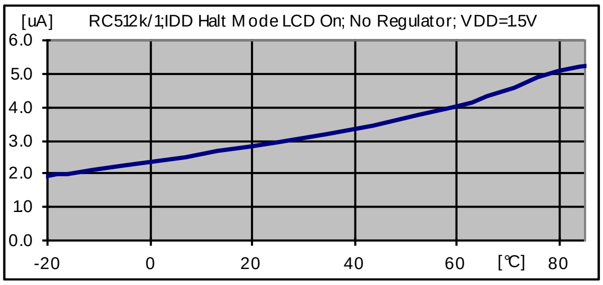


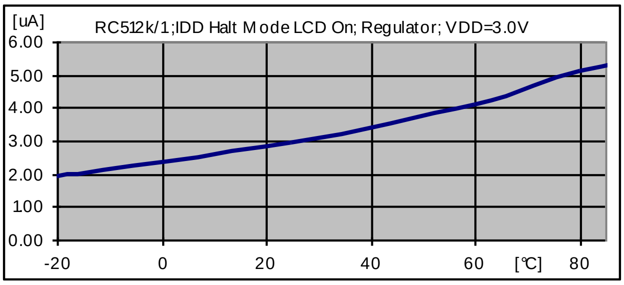


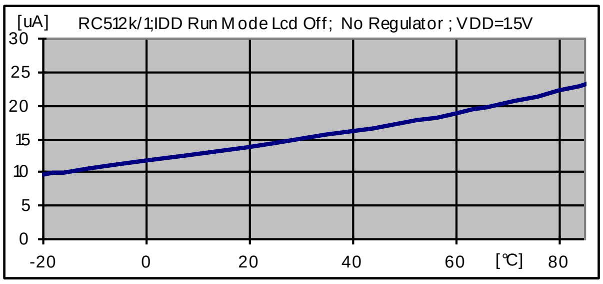
20.3 IDD Current (RC 512 kHz, CpuClk=512kHz/1)

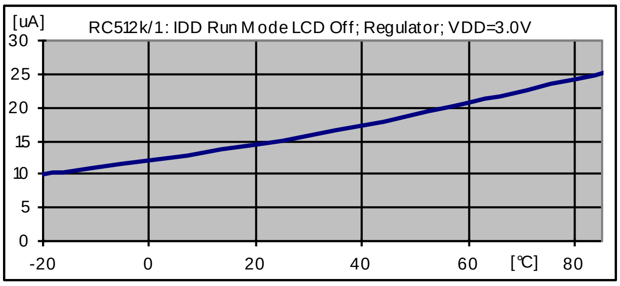


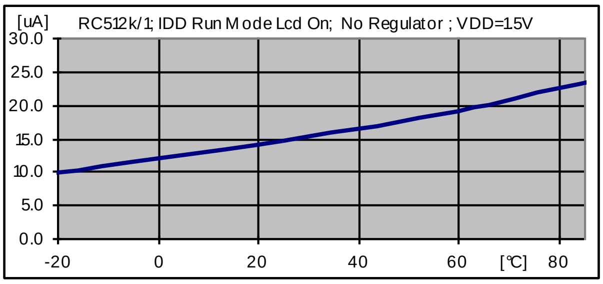


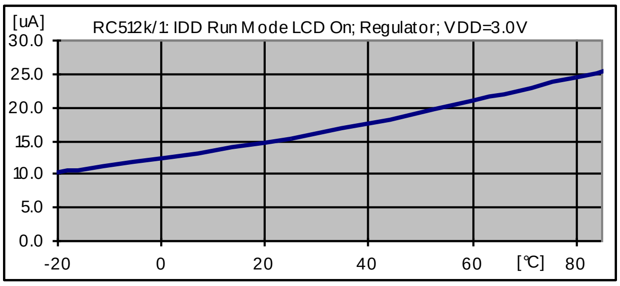


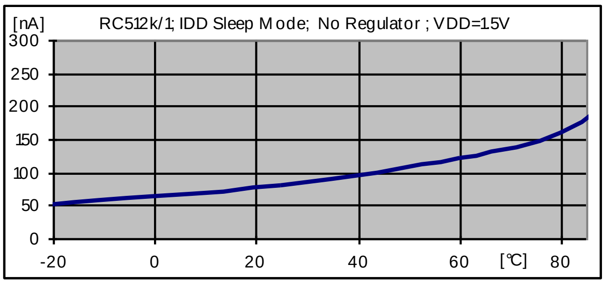


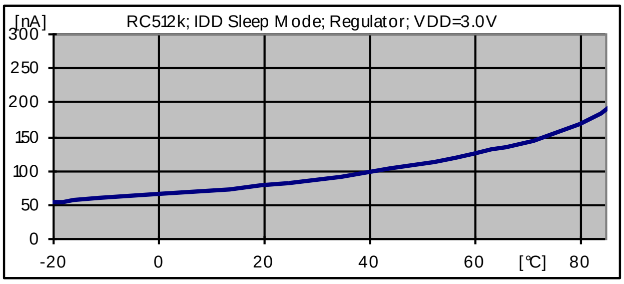




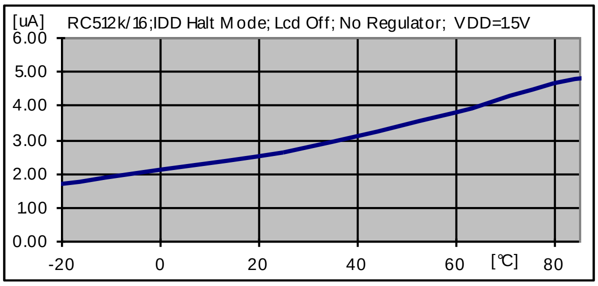


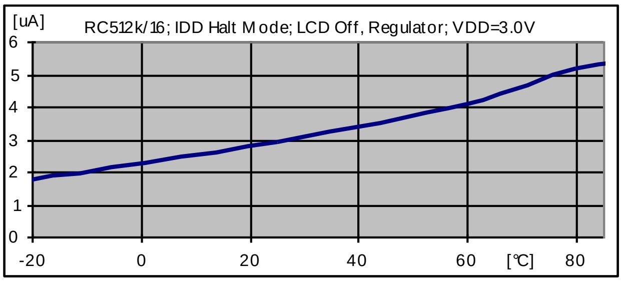


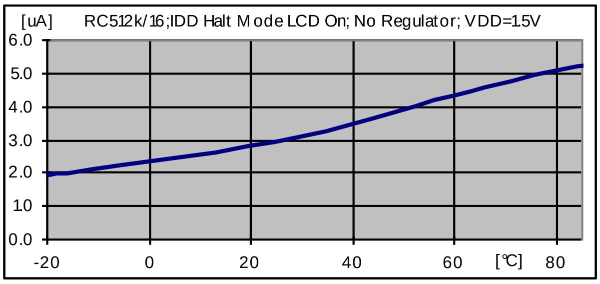


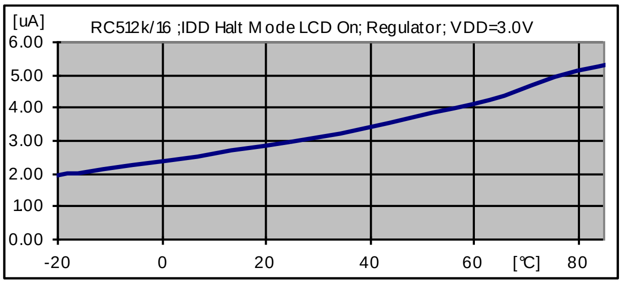


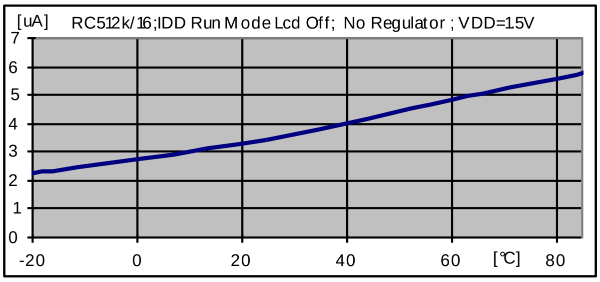
20.4 IDD Current (RC 512 kHz, CpuClk=512kHz/16)

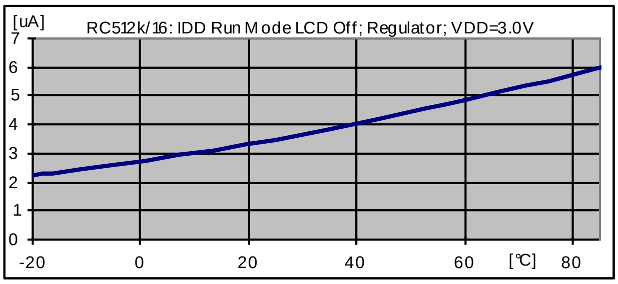


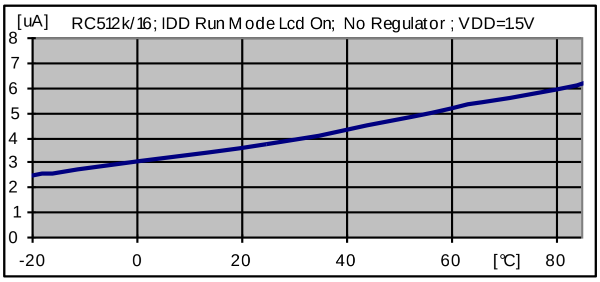


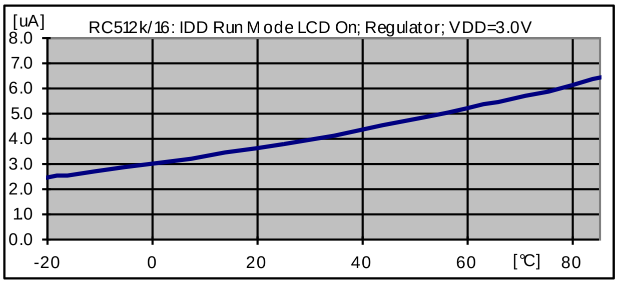


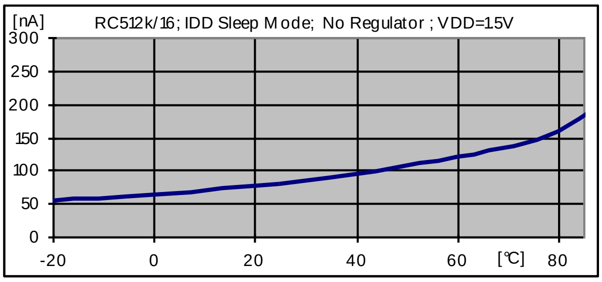


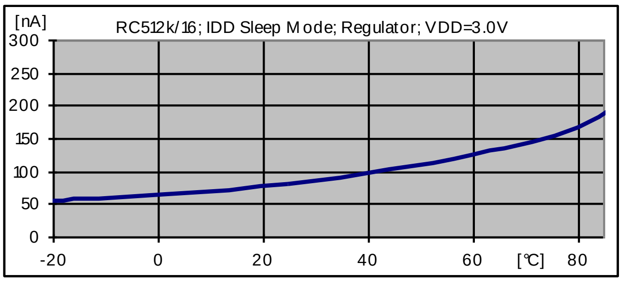












21. Electrical Specification

21.1 Absolute Maximum Ratings

	Min.	Max.	Units
Power supply V_{DD}-V_{SS}	- 0.2	+ 3.8	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	As per Jedec J-STD-020C		

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

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	-20	25	85	°C	
V_{DD}_Range	1.4	3.0	3.6	V	with internal voltage regulator
V_{DD}_Range2	1.2	1.5	1.8		without internal voltage regulator
V_{SS}		0		V	reference terminal
C_{VDDCA} (note 1)	100			nF	regulated voltage capacitor

Note 1: 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 Recommended Crystals

SMD-Series Crystal Fundamental Mode					
Nominal Frequency	F_L		32768		Hz
Serial resistance	R_S		65	85	kΩ
Motional capacitance	C_1		4.0		fF
Static capacitance	C_0		1.2		pF
Load capacitance	C_L	8		12	
Drive level	P			1.0	µW
DS-Series Crystal Fundamental Mode					
Nominal Frequency	F_L		32768		Hz
Serial resistance	R_S		35	60	kΩ
Motional capacitance	C_1		2.1		fF
Static capacitance	C_0		0.9		pF
Load capacitance	C_L		8.2		
Drive level	P			1.0	µW

21.5 DC Characteristics - Power Supply

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

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
ACTIVE Supply Current (in active mode with LCD on) No LCD display connected						
	RC512, 25°C, CPU 512kHz	$IVDD_{a512-512}$		15.2		uA
	RC512, T -20 ... 85°C, CPU 512kHz	$IVDD_{a512-512}$		15.2		uA
	RC512, 25°C, CPU 32kHz	$IVDD_{a512-32}$		3.8		uA
	RC128, 25°C, CPU 128kHz	$IVDD_{a128-128}$		3.8		uA
	RC128, -20 ... 85°C, CPU 128kHz	$IVDD_{a128-128}$		3.8		uA
	RC128, 25°C, CPU 32kHz	$IVDD_{a128-32}$		1.7		uA
STANDBY Supply Current (in Halt mode, LCDOff)						
	RC512, 25°C, CPU 512kHz	$IVDD_{h512-512}$		2.7		uA
	RC512, -20 ... 85°C, CPU 512kHz	$IVDD_{h512-512}$		2.7		uA
	RC512, 25°C, CPU 32kHz	$IVDD_{h512-32}$		2.7		uA
	RC128, 25°C, CPU 128kHz	$IVDD_{h128-128}$		0.75		uA
	RC128, -20 ... 85°C, CPU 128kHz	$IVDD_{h128-128}$		0.75		uA
	RC128, 25°C, CPU 32kHz	$IVDD_{h128-32}$		0.75		uA
LCD Multiplier current	25°C			0.3		uA
SLEEP Supply Current		IVDDs		0.1		uA
	-20 ... 85°C	IVDDs				uA
POR static level	-20 ... 85°C, No Load on Vreg	VPOR2		0.95		V
RAM data retention	-20 ... 85°C	Vrd2	1.2			V
Regulated voltage	Halt mode, No Load	Vreg		1.5		V

Note 2: LCD Display not connected.
Note 3: The instruction loop described in chapter 18 is used for these tests.

Supply Voltage Level Detector

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
SVLD voltage Level0	25C Trimmed value	V_{SVLD0}	1.30	1.38	1.46	V
SVLD voltage Level1	25°C Trimmed value	V_{SVLD1}	2.25	2.4	2.55	V
Adjust on Level 1 (up / down)		V_{SVLD0_adj}		50mV		
Adjust on Level 2 (up /down)		V_{SVLD1_adj}		200mV		
Temperature coefficient	0 to 50°C			+ 0.05		%/°C

21.6 RC Oscillator 512kHz

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

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
trimmed value	25 °C			512		kHz
Temperature stability	-20°C to 85°C,			4.7		kHz/°C
Voltage stability	With Vreg VDD=1.8 to 3.6V			0		kHz/V
Voltage stability	Without Vreg VDD=1.2V to 1.8V			11		kHz/V
RC trim step (LSB)	25°C, 3V			26		kHz/LSB

21.7 RC Oscillator 128kHz

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

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
trimmed value	25 °C			128		kHz
Temperature stability	-20°C to 85°C,			1.3		kHz/°C
Voltage stability	With Vreg VDD=1.8 to 3.6V			0		kHz/V
Voltage stability	Without Vreg VDD=1.2V to 1.8V			13		kHz/V
RC trim step (LSB)	25°C, 3V			6.5		kHz/LSB

21.8 DC characteristics - I/O Pins

Conditions: T= -20 ... 85°C (unless otherwise specified)
V_{DD}=1.5V
V_{DD}=3.0V

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
Input Low voltage						
Ports A,B,SP, Test, Reset	V_{DD} < 1.5V	V_{IL}	Vss		0.2V_{DD}	V
Ports A,B,SP ,Test, Reset	V_{DD} > 1.5V	V_{IL}	Vss		0.3V_{DD}	V
Input High voltage						
Ports A,B,SP ,Test, Reset		V_{IH}	0.7V_{DD}		V_{DD}	V
Output Low Current	V_{DD}=1.5V , VOL=0.15V	I_{OL}		1.1		mA
all logic outputs	V_{DD}=1.5V , VOL=0.30V	I_{OL}		2.1		mA
	V_{DD}=1.5V , VOL=0.50V	I_{OL}		3.1		mA
	V_{DD}=3.0V , VOL=0.15V	I_{OL}		1.9		mA
	V_{DD}=3.0V , VOL=0.30V	I_{OL}		3.9		mA
	V_{DD}=3.0V , VOL=0.50V	I_{OL}		6.4		mA
	V_{DD}=3.0V , VOL=1.00V	I_{OL}		12.0		mA
Output High Current	V_{DD}=1.5V, VOH= V_{DD}-0.15V	I_{OH}		-0.6		mA
all logic outputs	V_{DD}=1.5V, VOH= V_{DD}-0.30V	I_{OH}		-1.1		mA
	V_{DD}=1.5V, VOH= V_{DD}-0.50V	I_{OH}		-1.5		mA
	V_{DD}=3.0V, VOH= V_{DD}-0.15V	I_{OH}		-1.5		mA
	V_{DD}=3.0V , VOH= V_{DD}-0.30V	I_{OH}		-3.0		mA
	V_{DD}=3.0V , VOH= V_{DD}-0.50V	I_{OH}		-5.0		mA
	V_{DD}=3.0V , VOH= V_{DD}-1.00V	I_{OH}		-9.5		mA
Input Pull-down						
Test, Reset	V_{DD}=3.0V, Pin at 3.0V, 25°C	R_{PD}		20k		Ohm
Input Pull-down						
Port A,B,SP	V_{DD}=3.0V, Pin at 3.0V, 25°C	R_{PD}		100k		Ohm
Input Pull-up						
Port A,B,SP	V_{DD}=3.0V, Pin at 0.0V, 25°C	R_{PU}		100k		Ohm

21.9 LCD SEG[20:1] Outputs

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

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
Driver Impedance Level 0	Iout = ±5μA, Ext. Supply	R_{SEGVL0}			20	KOhm
Driver Impedance Level 1	Iout = ±5μA, Ext Supply	R_{SEGVL1}			20	KOhm
Driver Impedance Level 2	Iout = ±5μA, Ext Supply	R_{SEGVL2}			20	KOhm
Driver Impedance Level 3	Iout = ±5μA, Ext Supply	R_{SEGVL3}			20	KOhm

21.10 LCD Com[4:1] Outputs

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

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
Driver Impedance Level 0	Iout = ±5μA, Ext. Supply	R_{comVL0}			10	KOhm
Driver Impedance Level 1	Iout = ±5μA, Ext. Supply	R_{comVL1}			10	KOhm
Driver Impedance Level 2	Iout = ±5μA, Ext Supply	R_{comVL2}			10	KOhm
Driver Impedance Level 3	Iout = ±5μA, Ext Supply	R_{comVL3}			10	KOhm

21.11 DC Output Component

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

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
DC Output component	No Load	$\pm V_{DC\_com}$		20		mV

21.12 LCD Voltage Multiplier

Conditions: T=25°C, All Multiplier Capacitors 100nF, freq=512Hz. (unless otherwise specified)

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
Voltage Bias Level 1	1 uA load, level 7	V_{VL1}	0.95	1.05	1.18	V
VL1 selection range (level 0 to 15)				Refer to Figure 36		
Voltage Bias Level 2	1 uA load	V_{VL2}		2.10		V
Voltage Bias Level 3	1 uA load	V_{VL3}		3.15		V
Temp dependency V_{VL1}	1 uA load, -10...60°C	dV_{VL1}/dT		-4.6		mV/°C

V_{VL2} = 2 x V_{VL1}, V_{VL3}=3 x V_{VL1}

21.13 EEPROM

Parameter	Conditions	Symb.	Min.	Typ.	Max.	Unit
Read time (note 9)	-20 ... 85°C	EEPrd		45		us
Write time (note 9)	-20 ... 85°C	EEPwr		15		ms
VDD during read operation	-20 ... 85°C	V_{EEPRd}	VDDmin		VDDmax	V
VDD before and during write operation	-20 … 85°C	V_{EEPWd}	V_{SVLD0}		VDDmax	

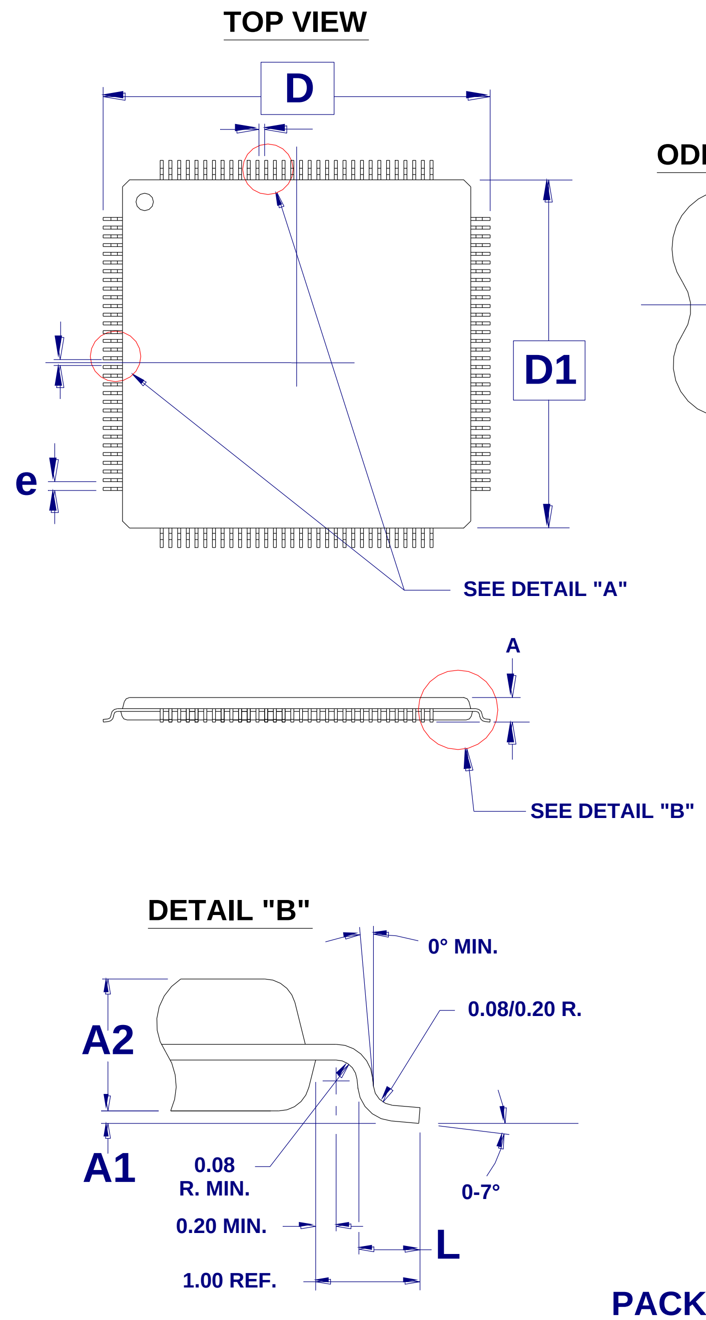
Note 9 : The Read write typical values are based on system clock divisions and are guaranteed by design

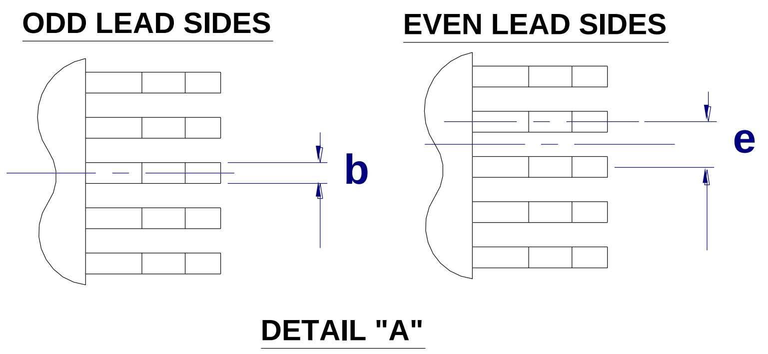
22. Die, Pad Location and Size

Available on specific request

23. Package & Ordering information

23.1 TQFP-52





SYMBOL	TQFP52 ALL DIMENSIONS IN MILLIMETERS		
	MIN.	TYP.	MAX.
A	~	~	1.60
A_1	0.05	~	0.15
A_2	1.35	1.4	1.45
D	12.00 BSC.		
D_1	10.00 BSC.		
L	0.45	0.60	0.75
N	52		
e	0.65 BSC		
b	0.22	0.32	0.38

1.00/0.10 MM FORM, 1.4 MM THICK

PACKAGE OUTLINE, TQFP, 10X10 MM BODY,

23.2 Ordering Information

Packaged Device:

EM6627 TQ52 B - %%%

Package:
TQ52 = TQFP 52 pin

Delivery Form:
B = Tape & Reel
D = Trays (Plate)

Customer Version:
customer-specific number
given by EM Microelectronic

Device in DIE Form:

EM6627 WS 11 - %%%

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

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

Customer Version:
customer-specific number
given by EM Microelectronic

Ordering Part Number (selected examples)

Part Number	Package / Die Form	Delivery Form / Thickness
EM6627TQ52B-%%%	TQFP 52	Tape & Reel
EM6627TQ52D-%%%	TQFP 52	Trays (Plate)
EM6627WS11-%%%	Sawn Wafer	11 mils
EM6627WP11-%%%	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 numbers %%% (e.g. 005 , 012, 131, etc.).

23.3 Package Marking

TQFP52 marking:

First line:	E	M	6	6	2	7		%	%	%	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

6	6	2	7	%
%	%	P	P	
P	P	P	P	P

Where: %%% = customer version, specific number given by EM (e.g. 005, 012, 131, 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

23.4 Customer Marking

There are **11** digits available for customer marking on **TQFP52**

Please specify below the desired customer marking (TQFP52 only).

24. Product Support

Check our web site under Products/RF Identification section. Questions can be sent to info@emmicroelectronic.com.