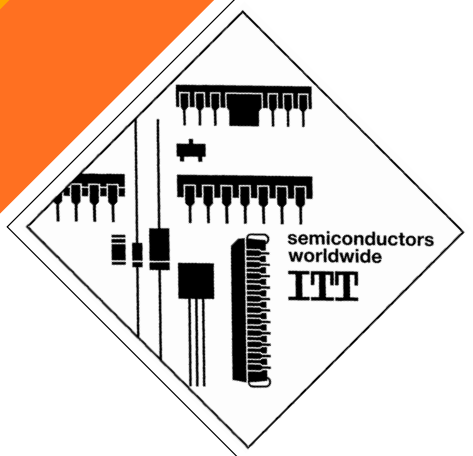


CCU 3000,
CCU 3000-I,
CCU 3001,
CCU 3001-I,
Central Control Unit

Edition Feb. 14, 1995
6251-367-1DS

ITT Semiconductors

Contents

1. Introduction

The CCU 3000, CCU 3000-I, CCU 3001, CCU 3001-I are integrated circuits designed in 1.2 µm CMOS technology, with the exception of CCU 3000, TC18 and TC19, which is designed in 1 µm CMOS technology. The CPU contained on the chips is a functionally unchanged 65C02-core, which means that for program development, systems can be used which are on the market; including high level language compilers.

The pin numbers mentioned in this data sheet refer to the 68-pin PLCC package unless otherwise designated.

The CCU 3000-I is described separately in an addendum on page 66.

1.1. Features of the CCU 3000, CCU 3000-I, CCU 3001, CCU 3001-I

– CCU 3000 = ROM-less version of the CCU 3001

– 65C02 CPU with max. 8 MHz clock

– 32 kByte internal ROM (CCU 3001 only)

– 1344 internal Bytes RAM with stand-by option

– 51 I/O lines (CCU 3001)

– 26 I/O lines (CCU 3000)

– clock generator with programmable clock frequency

– 8 level interrupt controller

– CCU 3000, CCU 3001:
2 Multimaster IM bus interfaces

– CCU 3000-I, CCU 3001-I: 1I^2C/IM bus and
1 Multimaster IM bus interface (see addendum)

– IR-input for software-decoded IR-systems

– on-chip power on, stand-by and clock supervision logic

– on-chip watchdog

– 3 multifunctional timers

– supports memory banking (external 2MBytes)

– power down signal for external memory

– mask option: EMU mode

– programs can be written in Assembler or in “C”

– CCU 3000 TC 18/19: 1.0 µm CMOS technology, (see addendum)

– application software available

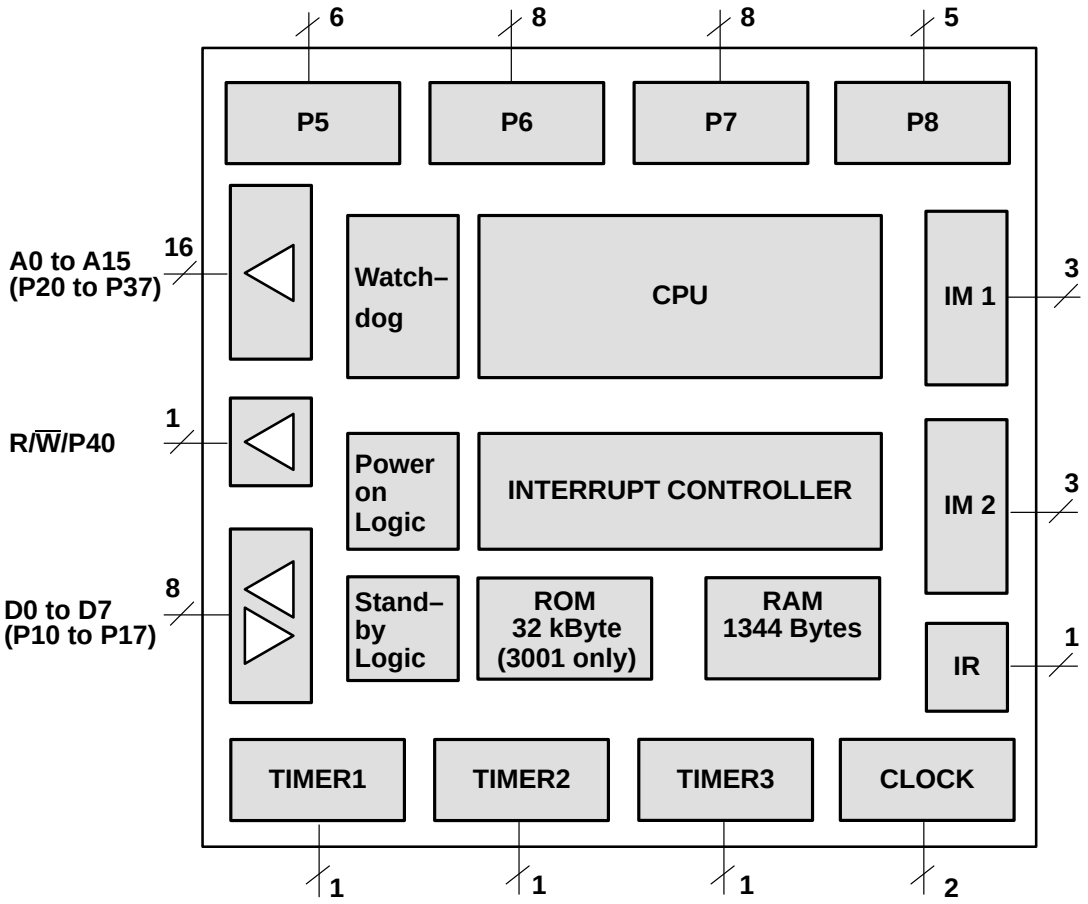


Fig. 1–1: CCU 3000, CCU 3001 block diagram

2. Functional Description

2.1. ROM

The chip is equipped with 32 kByte mask-programmable ROM. The ROM uses up the address space from 8000H to FFFFH. This ROM can be supplemented or replaced externally. Only the CCU 3001 has an internal ROM.

2.2. RAM

The RAM area is split into three parts:

– page 0 (address 0 to FFH)

– page 1 (address 100H to 1FFH)

– page 3, 4, 5, 6 (address 300H to 63FH)

Page 0 offers a particularly fast access to the 65C02 and is therefore very valuable for fast, compact programs. Page 1 contains the stack and must therefore also have RAM. The remaining RAM-memory follows in pages 3, 4, 5, 6, as page 2 is reserved as I/O address space. The RAM can be kept in the stand-by mode via stand-by pin.

2.3. CPU

The CPU core is fully compatible with the 65C02 microprocessor. However, not all the pins of the 65C02 processor are accessible for the user outside the chip. One switch in the control register allows the CPU to be switched off, so that an external processor can take over its tasks. This external processor can of course also be an in-circuit emulator, which makes near-hardware emulation possible, even though the status and control lines of the internal CPU are not accessible. If an external processor is used, all hardware blocks of the chip are as accessible to it as if it were the internal CPU.

2.4. Clock Generator

An integrated two-pin oscillator generates the clock for the microcontroller. The frequency created by the oscillator can be programmed to be reduced with a divider by the factor 1 ... 255. This enables the user to decrease the current consumption by the controller by reducing the working frequency as well as to increase the access time for the (slower) external memory. This divider contains the value 4 after a reset, so that the system can also start with a slow external memory. If the mask-option OSC is set (EMU version), a switch in the control register makes it possible to receive the internal clock $\Phi 2$ at XTAL2. In this case the oscillator must be external and the clock must be fed to the pin XTAL1. In this way, the user gets a time reference for internal operations in the microcomputer. This is especially important with the interrupt controller. **The production version of the CCU does not have this function!**

2.5. PORT 1 to PORT 3, PORT 6 to PORT 8

8 ports belong to the system, of which 5 are 8 bits wide, one 6 bit, one 4 bit and one 1 bit wide. All port lines of PORTS 1 to 3 and 6 to 8 can be used as inputs or outputs independently from each other. One register per port defines the direction. PORT1 to PORT3 have push-pull outputs and PORT6 to PORT8 have open drain outputs. Even a line defined as output can be read, the pin level being important. This property makes it possible for the software to find desired and undesired short circuits. Each port reserves a byte for the direction register and the data in the I/O page. If the corresponding bit in the direction register is set to 0, the output mode is switched on. After a reset, all bits of a direction register are set to 1. The falling edge of bit 7 of PORT 8 generates interrupts if the priority of the corresponding interrupt controller source (7) is not set to 0.

2.6. PORT 4

PORT 4 consists of only one line (LSB, P40). After a reset, PORT 4 operates as an input only. As soon as PORT 4 is written for the first time, it is switched to output mode (push-pull). Later read accesses read the actual level at port 4. If bit 3 in the control word is active, P4 is used as an R/$\overline{W}$-line. If the internal CPU is active, R/$\overline{W}$ is an output line, otherwise it is an input. But P4 has another, very important function during RESET. The level at P4 during RESET decides whether the control word is read from the internal ROM (FFF9H) or from the external memory. *It is therefore important that the desired level during RESET is set at P4.* An internal pull-down resistor of approx. 100 kΩ is integrated in the CCU 3001, which ensures that the control word is read by the internal ROM. The external control word access is obtained via an external pull-up resistor of approx. 5 kΩ. The CCU 3000 has an internal pull-up resistor at P4 (external ROM access). The further mode of operation of the CCU 3000, CCU 3001 depends **only** on the control word though.

Please note that this mode is always necessary for the CCU 3000 since this device does not have internal ROM!

2.7. I/O-Lines P50 to P55

The 6 additional I/O-lines have a two-fold function:

– input or output line (open drain output) or

– fully decoded I/O-select lines (push-pull outputs)

As a rule these lines can be used as input or output lines. As soon as ports 1 to 4 are used as system bus, they are lost as I/O-channels. However, a total of 48 port lines (24 inputs and outputs each) can be reconstructed without difficulties (1 housing for 8 lines), if the additional 6 I/O-lines of the CCU 3000, CCU 3001 are switched into the port select mode. They then represent the select lines of the original ports 1 to 3. Each line can be defined as I/O or port select line separately. In the I/O-page three bytes are needed.

Mode 1 bit 5,4,3,2,1,0 Switch I/O – port select mode
0... I/O line (default)
1... port select

Direction 2 bit 5,4,3,2,1,0 direction switch for I/O lines 5 ... 0
1... input (default)
0... output

Data 3
bit 5 I/O 5 or RDPort 1
bit 4 I/O 4 or WRPort 1
bit 3 I/O 3 or RDPort 2
bit 2 I/O 2 or WRPort 2
bit 1 I/O 1 or RDPort 3
bit 0 I/O 0 or WRPort 3

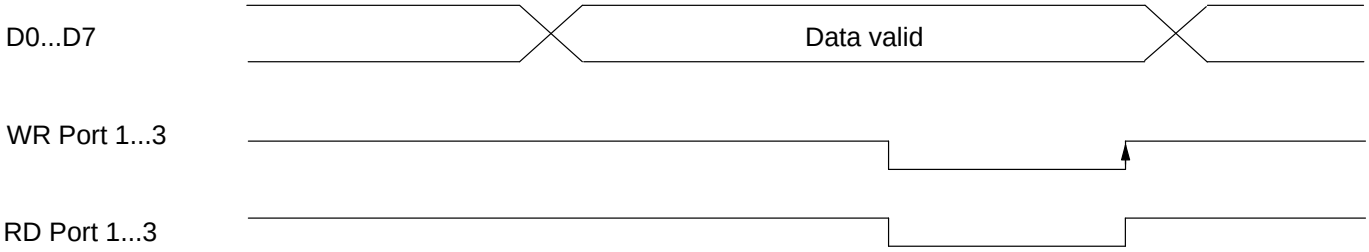


Fig. 2–1: Timing diagram

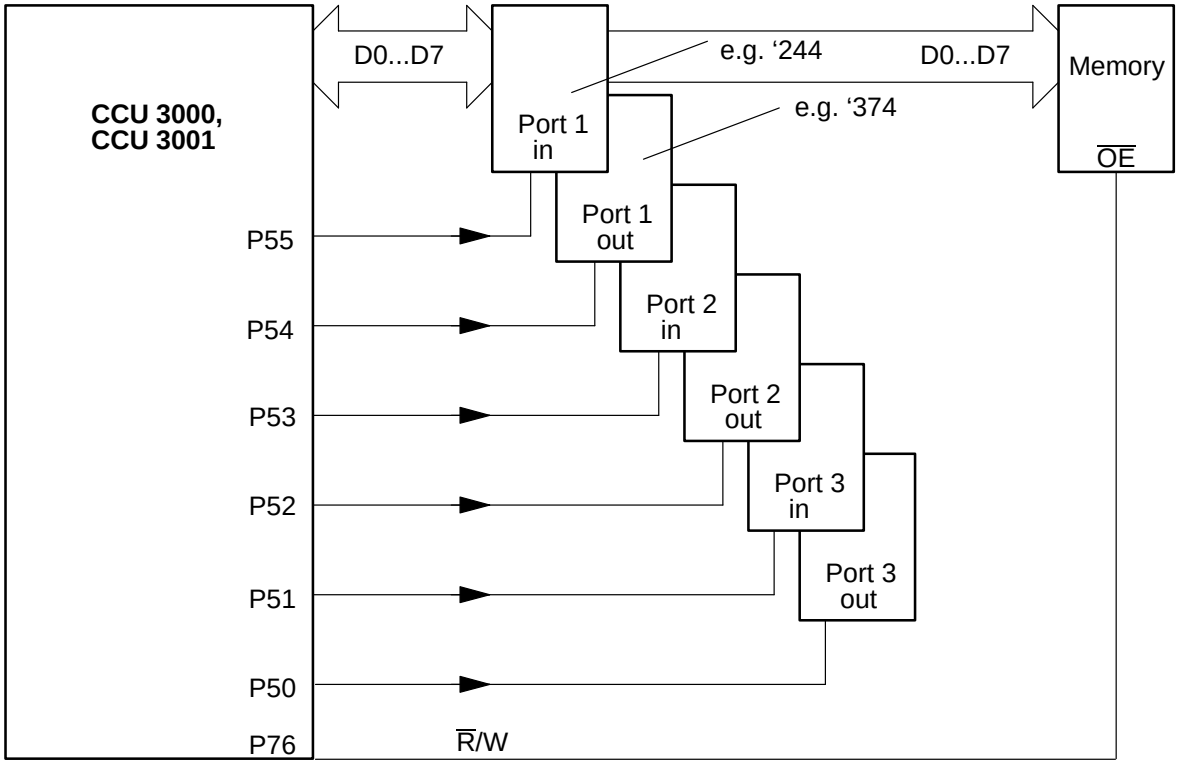


Fig. 2–2: External reproduction of ports 1 to 3

2.8. Special Mode of Port 7

Each line of port 7 can be switched independently into a special mode. This mode is selected by the mode control register. After reset this register is set to 0 (= Port mode). A “1” in this register turns this line into the special mode. As the control signals are all outputs, the direction for those lines must be defined as outputs (reset condition = inputs)

Special Mode of	Function
P77	Power-down control for ext. memory
P76	R/W output
P75 ...P70	6 bit banking addresses with common home bank logic

All special mode signals have push-pull outputs. (Port mode: open drain).

2.8.1. Power-down Control External Memory (Special Mode P77)

In many applications the power consumption of the controller should be reduced when the system goes into standby mode. The programmable clock of the CCU allows this, but external memories do not automatically reduce their power consumptions when the access speed is slower. These devices need a separate control signal for power down. Special out of P77 delivers such a signal. It is low for the last two XTAL cycles before, and 0.5 cycles after the rising edge of the internal PHI2 clock. This guarantees a wake-up and address time of 2 cycles and a maximum active time of 2.5 clock cycles for each PHI2 period. At higher speeds the P77 special out stays low.

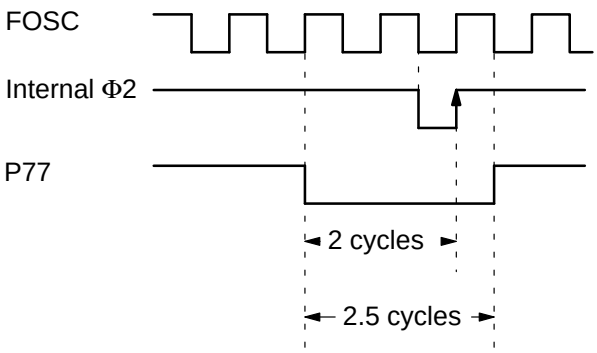


Fig. 2–3: Power-down control

Please note that during and after reset P77 is a port line (= tristate) until the special mode and the direction register is set by software. A pull-down resistor on the power-down input of the memory is necessary to allow the CCU the access to the control word and the first instructions.

2.8.2. $\overline{R}$/W Output (Special Mode P76)

This is the negated R/$\overline{W}$-line of the CPU. Can be used for $\overline{CE}$ or $\overline{OE}$ control on memories. With a pull-down resistor on this pin it is active during RESET.

2.8.3. Banking Address (Special Mode P70 to P 75)

Banking is done in 32 KByte banks. The first bank (000H to 7FFFH) includes the RAM, the I/O-page and ROM (all other locations) and is used as a home-bank for the banking controller, interrupt routines, common subroutines etc. The second half of the address space (8000H to FFFFH) is banked.

In the special mode of Port 70 to 75 the content of the data register is output as long as the address A15=1.

A low level of A15 forces all special outs of P70 to P75 to become '0'. The data register can contain the bank addresses 1 to 63. This bank is used for CPU accesses from 8000H to FFFFH. Low accesses are always done to bank 0, independent of the data of Port 7.

Note:

– all upper banks must contain the interrupt vectors. Bank 0 must have the control word and reset vector.

– during and after reset P7 is in the Tristate-Port-Mode. To make sure that the control word and the reset vector can be accessed use high impedance pull-down resistors on all special-out P7 lines. The control word and the reset vector are then accessed out from bank 0. The init routine (where P7 will be defined as special out) must be in bank 0.

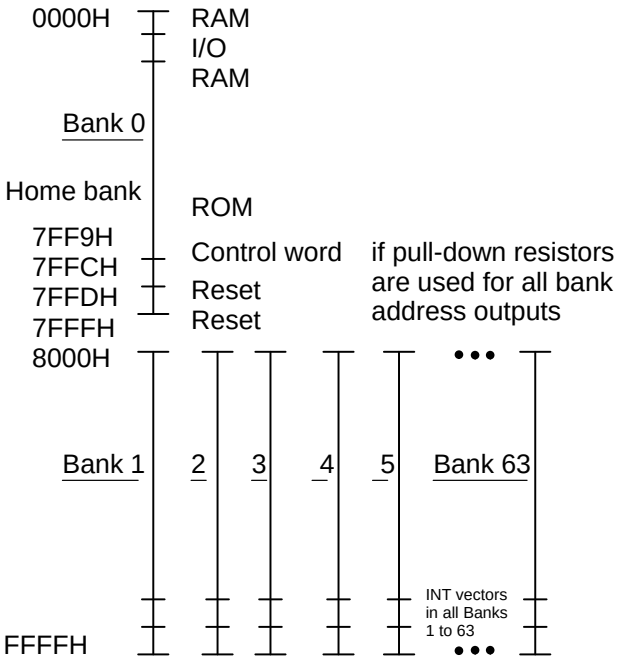


Fig. 2–4: Memory Map, up to 2 MByte

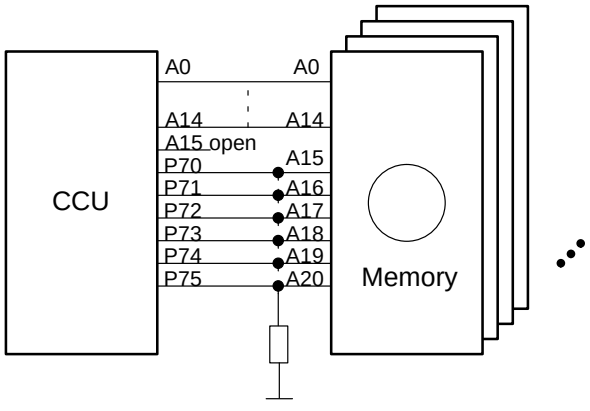


Fig. 2–5: Banking with 32 kByte banks

2.9. Reset Function

The internal reset provides a correct basic setup of the complete hardware on the chip. For this the internal control register is loaded during reset. One reserved byte in the ROM is accessed by the reset circuit and its content is copied into the control register. The internal voltage supervision resets the IC if the voltage is too low. The reset pin is also used as output for internal reset sources (watchdog, power-down detector, clock supervisor). Internal resistors limit the maximum current.

2.10. Control Register (address 201H)

This is a combination of control switches in an 8-bit register. During reset it is loaded with the contents of the address FFF9H, but it can also be read and written via software. The controller starts operation with the setting dictated by reset. The switches have the following functions:

bit 0	CPU_disable	(low active)
bit 1	RAM_disable	(low active)
bit 2	ROM_disable	(low active)
bit 3	R/$\overline{W}$-mode P4	(low active)
bit 4	Bus external	(low active)
bit 5 to 7	set to 1	

The setting at the R/$\overline{W}$-pin decides whether the control word is read internally or externally. Bit 0 to bit 2 are the switches which can disable RAM, ROM and CPU. For external access a pull-up resistor must be connected to the R/$\overline{W}$ pin (CCU 3001). Bit 4 switches P1, P2 and P3 into the system bus mode. If the internal CPU is active, the direction of the data bus drivers is automatically set correctly, so that no additional decoding is necessary. Bit 3 switches P4 into the R/$\overline{W}$ mode. If no external write access is necessary, (ext. EPROM), P4 can stay in the port mode.

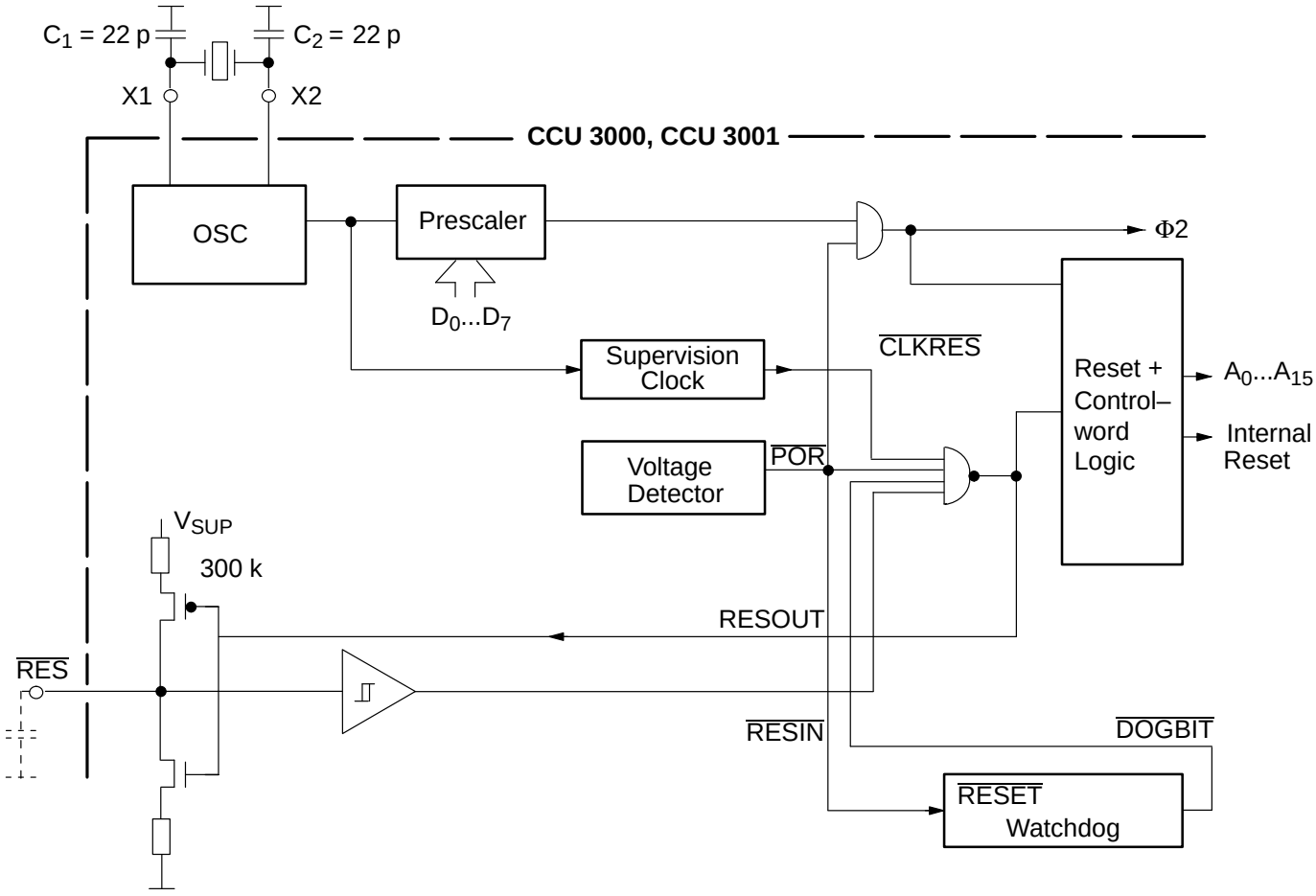


Fig. 2–6: Oscillator and reset

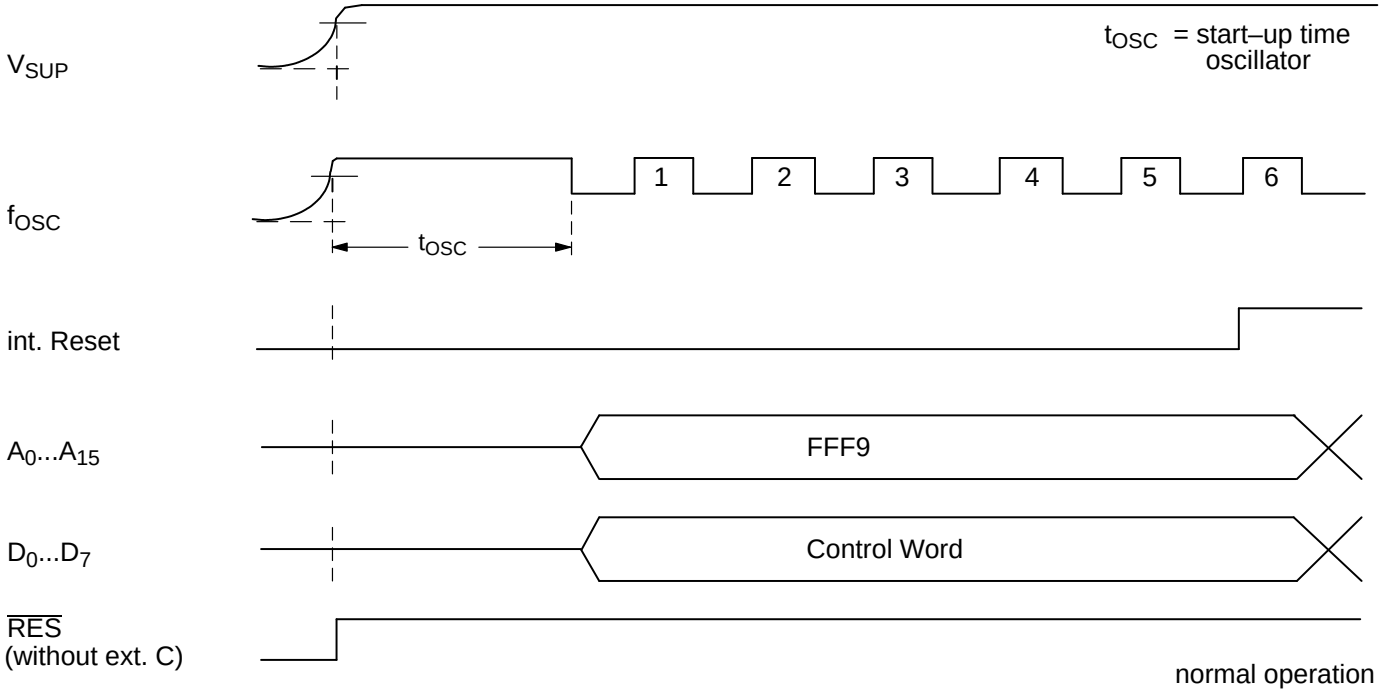


Fig. 2–7: Power-on sequence

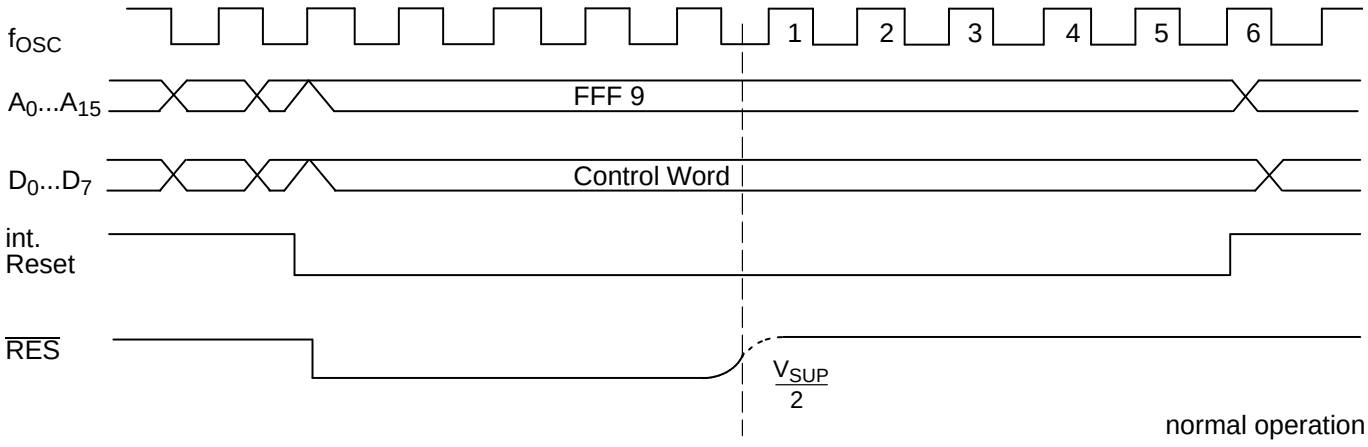


Fig. 2–8: External reset sequence

2.11. Interrupt Controller

The most important properties of this controller are:

– 8 sources

– 8 freely programmable priorities for every source

– maximum delay of 3 clock cycles

– vectorized interrupts, i.e., automatically the correct routine is accessed

– also to be used for external CPU

– option: disable after interrupt (resettable by software)

Running service routines are only interrupted if interrupts are enabled and a request of higher priority arrives. All others are stored and executed when interrupts of higher priority have been finished. Priority 0 means that the corresponding interrupt is disabled. (Priorities 1-7 lead to interrupts). One property of the controller is that the CPU is not modified, but vectorization takes place all the same. Thus the use of this controller is also possible for external CPUs (emulator!).

Solely the return from a service program differs slightly in software from the methods normally used for the 65C02. *The last command before the "RTI" must be a write operation into the return register of the controller.* This tells the controller that the service routine has been completed. Apart from this return register the controller occupies further 5 bytes. One of these serves as a control byte, the others incorporate the priorities for 8 sources. The controller therefore needs 6 bytes of the I/O-page. The control byte comprises:

bit 0	CLEAR_ALL_REQUESTS	(low active)
bit 1	ALLOW_ONE_INTERRUPT	(low active)
bit 2	DISABLE_INTERRUPTS	(low active)
bit 3	DISABLE_AFTER_INT	(low active)
bit 4	RESET_CONTROLLER	(low active)

All bits reset to 1 (inactive).

CLEAR_ALL_REQUESTS clears all interrupt flags at the same time. ALLOW_ONE_INTERRUPT is used in connection with the DISABLE_AFTER_INT (bit 3), to allow access to the next interrupt. DISABLE_INTERRUPT does not allow any interrupts, the request flags are set however. With the exception of bits 3 and 2 (DISABLE_INTERRUPTS, DISABLE_AFTER_INT) these are all dynamic signals, that is, the write process itself sends an appropriate signal. This has the same impulse length as the 65C02. Each of the 4 priority registers contains the priorities for 2 interrupt sources.

bit 7, 6, 5, 4	Priority for sources 8, 6, 4, 2
bit 3, 2, 1, 0	Priority for sources 7, 5, 3, 1

To connect an external CPU (emulator) with the controller, only two ICs of the 74-family are needed.

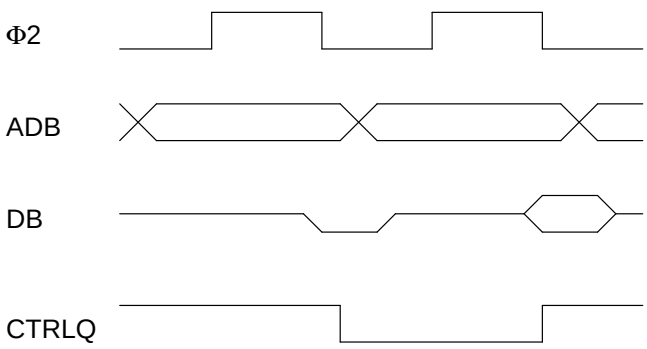


Fig. 2–9: Dynamic control signals interrupt
Bits 0, 1, 4

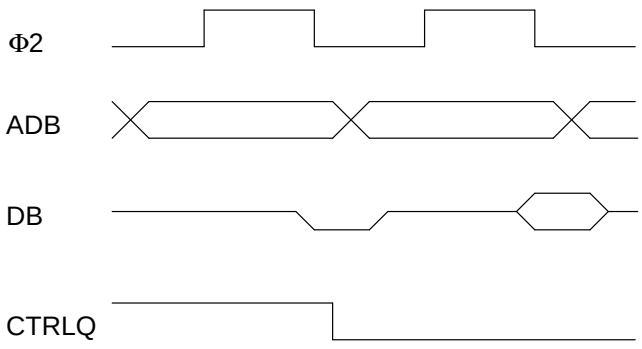


Fig. 2–10: Static control signals interrupt
Bits 2, 3

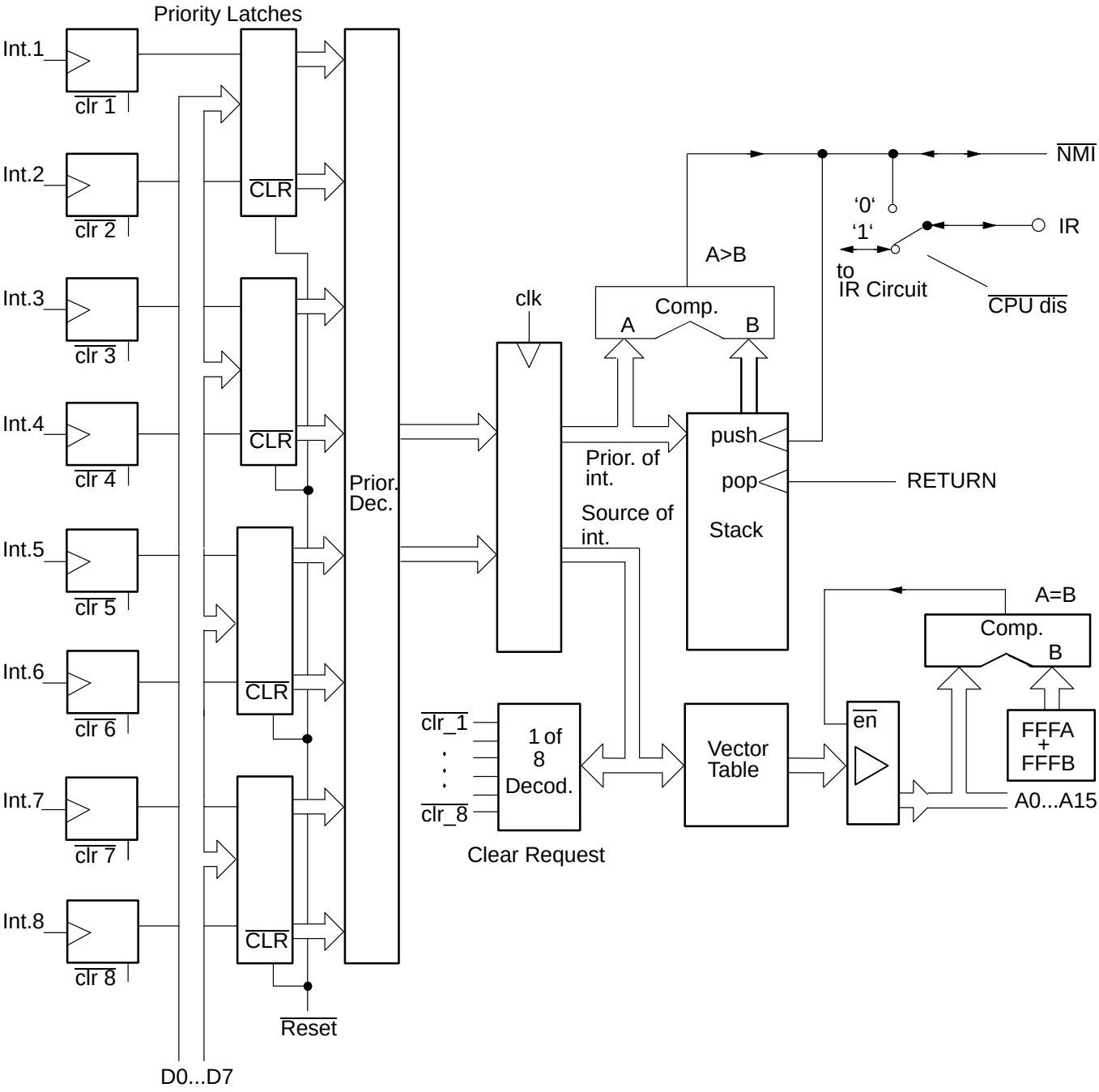


Fig. 2–11: Interrupt controller

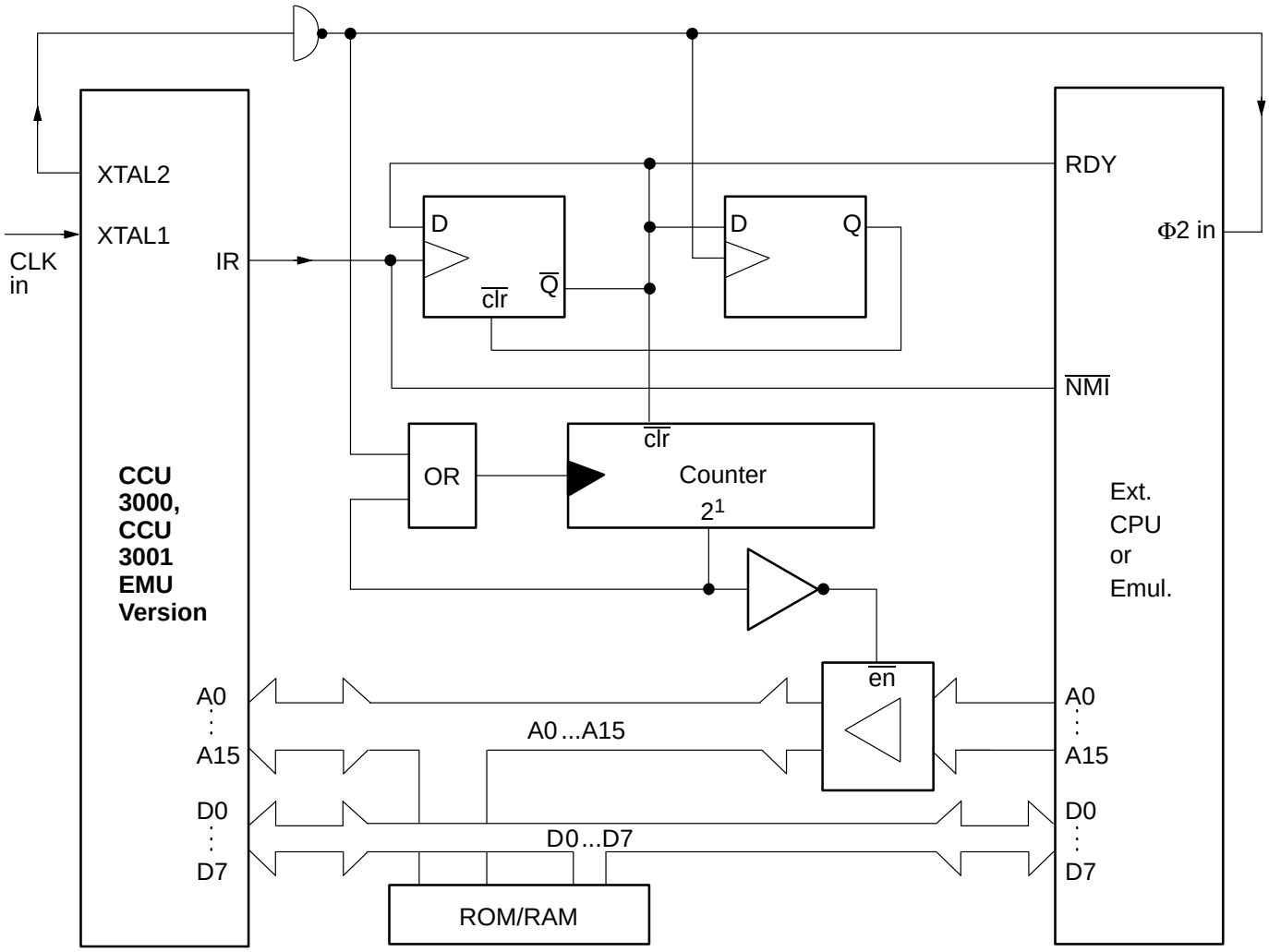


Fig. 2–12: Using an external CPU

2.12. IM Bus Interface

The IM bus has been improved in its characteristics for the CCU 3000, CCU 3001. In comparison to the interface of the CCU 2000 series it differs in:

– multimaster ability

– 3 slave registers (8 bit wide)

– higher speed possible

The multimaster ability permits the use of several CCUs on the same IM bus without impeding each other. Specially in add-on systems or systems with need of high computing power and/or I/O requirements, this offers great advantages. If several CCUs are admitted in a system, it must be ascertained that these can communicate with each other. A slave IM bus interface has been installed for this purpose. Parallel to the lines of the master, three completely independent receiver registers have been installed. All of these are constantly alert, whether the master itself is active or not. As all CCUs have the same IM bus addresses for these registers, the contents of these registers (that is, for all CCUs that are in the system) will be the same. The handshake amongst these is realized in software, and one register each is reserved for the device address, the request and the data to be transported. The data rate can now be adjusted per software. It is possible to attain 1 MBit/s, if the bus participants in question are devised to support this rate. Also the actual realization of the bus can forbid such a high data rate. The IM bus interface needs external pull-up resistors.

In the I/O-page the IM bus interface reserves 8 bytes:

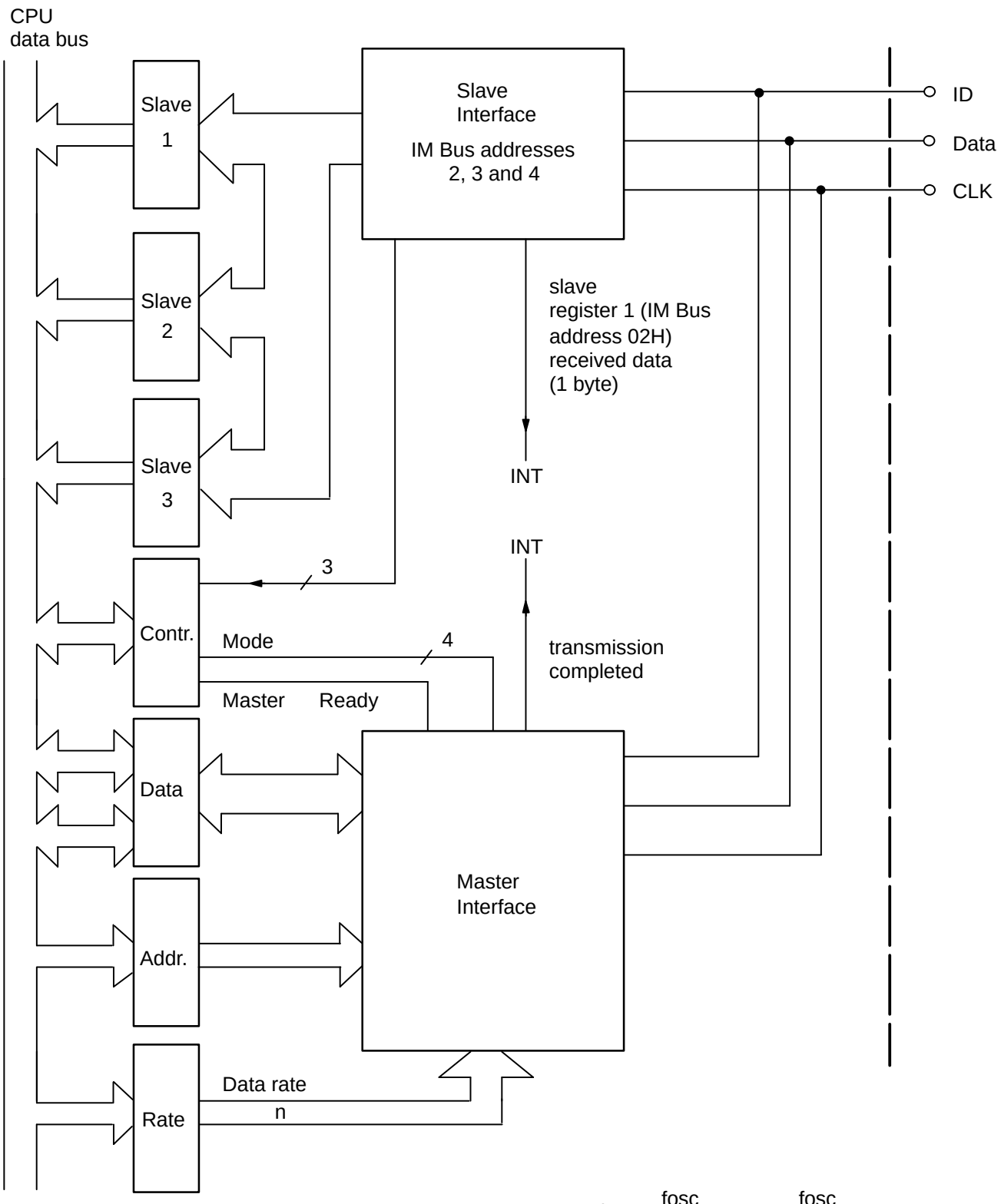
3 bytes		slave receiving registers (read)
1 byte		master address (write)
2 bytes		master data register (read/write)
1 byte		control register (read/write)
read:	bit 0	0... IM bus master ready 1... IM bus master busy
	bit 1	1 Byte received in slave register 1 (may generate interrupt)
	bit 2	IM-bus 1 control and status (may generate interrupt)
	bit 3	Word 3 Received (may generate interrupt)

write: bit 0 1... Write 8 bit
bit 1 1... Write 16 bit
bit 2 1... Read 8 bit
bit 3 1... Read 16 bit

1 byte Data rate (5 bits) = $\frac{fosc}{4 \cdot n}$

Only one of the bits 0 to 3 in the control register should be set. If all bits are set to '0', a reset of the interface is done, thus deleting telegrams still waiting for access to the bus. The IM bus addresses for the slave registers are:

Slave register	IM bus address
1	02 H
2	03 H
3	04 H



Data rate: $f_{IM} = \frac{fosc}{4 \cdot n}$; $n = \frac{fosc}{4 \cdot f_{IM}}$

Fig. 2–13: IM bus interface

2.13. Multifunctional Timer

The multifunctional timer for the CCU 3000, CCU 3001 has quite an unusual structure. It can serve as:

– event counter

– frequency counter

– pulse-length meter

– timer

– rate multiplier

– PWM

– asynchronous, serial interface

Each timer has a reserved pin and an interrupt. The pin is either input or output, depending on its function. Used as an output it has a push-pull structure. The timer consists of three main parts:

– start and stop detector

– internal time reference

– accumulator and arithmetic unit

The start and stop detector controls the internal pulse generator to synchronize counter and meter operations. The timer itself does not consist of a counter circuit, but of an accumulator and an adder. This configuration works as a counter with adjustable step length, as a shift register, as a PWM and as a rate multiplier. Change-over of operation modes can easily be effected.

Each of the multifunctional timer circuits of the CCU is realized as two 8-bit accumulators. In addition, there is a separate adder register for each of them. Both the accumulator and the adder may be accessed by the CPU via data bus. The accumulator has a shadow register the CPU may write to and the adder bus register may be read and written to.

While the adder register forms one side of the adder, the other side is either the output of the adder or the content of the accu shadow register. With every accu clock pulse either of these bytes is used. If no “LOAD” signal is active, the adder output is used. With “LOAD” active, the following accu clock pulse uses the content of the accu shadow register as adder input. The “LOAD” signal is derived from 1 out of 4 sources, selectable with bits 3 and 4 of control byte 3. Accu clock is selectable with bits 1 and 2 in control byte 2. Instead of the content of the adder register, accessible by the CPU, a hard-wired ‘-1’ may be used as input of the other side of the adder (bits 1 and 2 in control byte 3). By adding ‘-1’ to the accu’s content, the adder works as a standard down counter. With specific “READ LATCH” signals (control byte 2, bits 3, 4 and 5) and using the adder register as adder input, its content defines the step width of the counter.

In addition to its parallel byte connections, the adder registers have serial inputs and outputs. A serial clock shifts its contents. To hit the middle of serial data, the timer’s prescaler has a half load feature, controlled with bit 1 in control byte 1.

Fig. 2–14 shows a detailed diagram of the high part of the reloadable accumulator and its adder register. For examples of timer applications please refer to “Application Note CCU 3000/3001 Timers”.

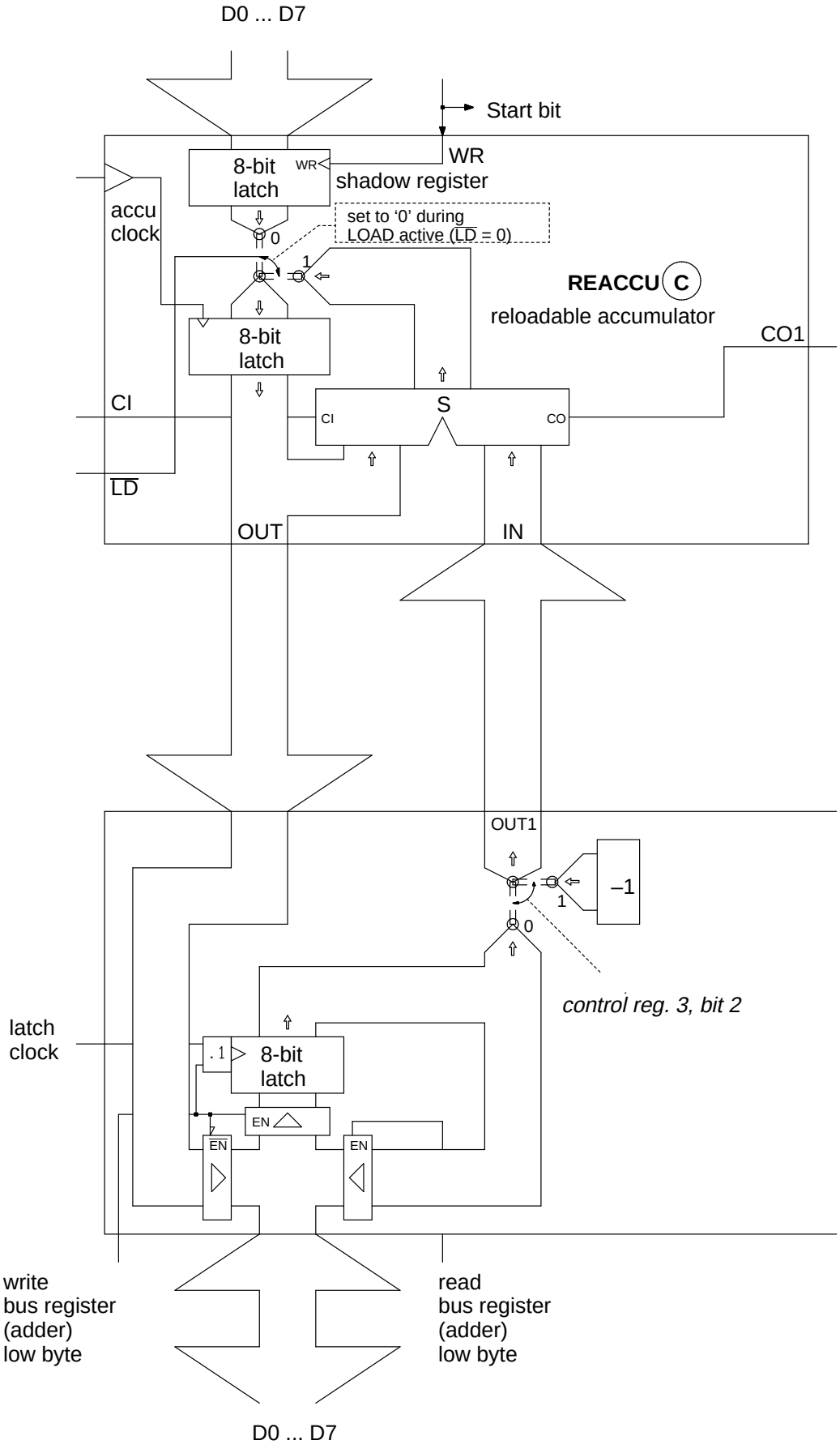


Fig. 2–14: Reloadable accumulator

Apart from the start values for the counter and adder registers, three control registers shift the timer into the preselected function.

Registers to control the timer:

2 bytes	prescaler high and low byte (read/write) (read: prescaler) (write: prescaler and reload register)
2 bytes	accu high and low byte (write) (write: accu and reload register)
2 bytes	adder register high and low byte (read/write)
3 bytes	control register 1 to 3 (write)

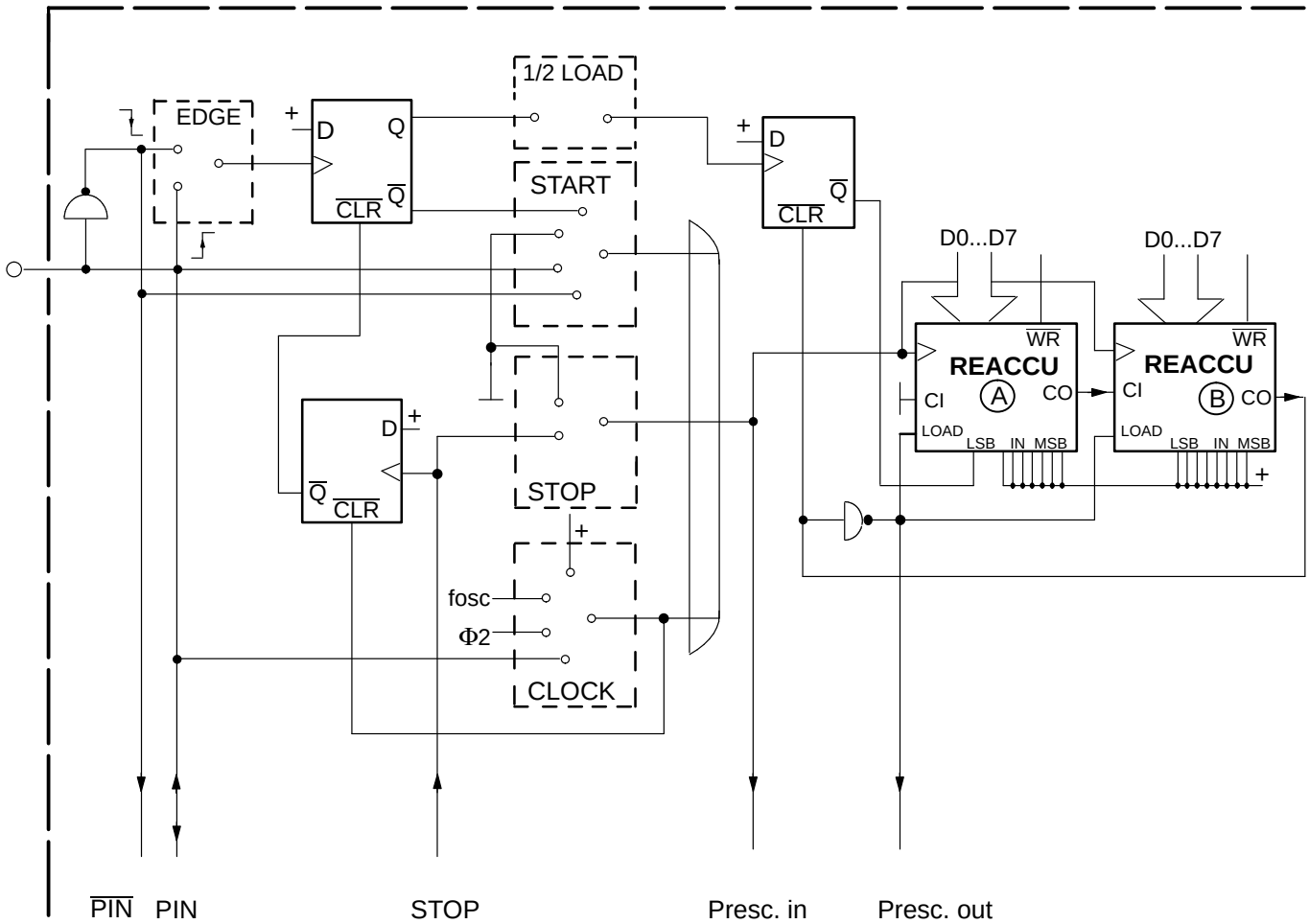


Fig. 2–15: Prescaler timer, start/stop logic

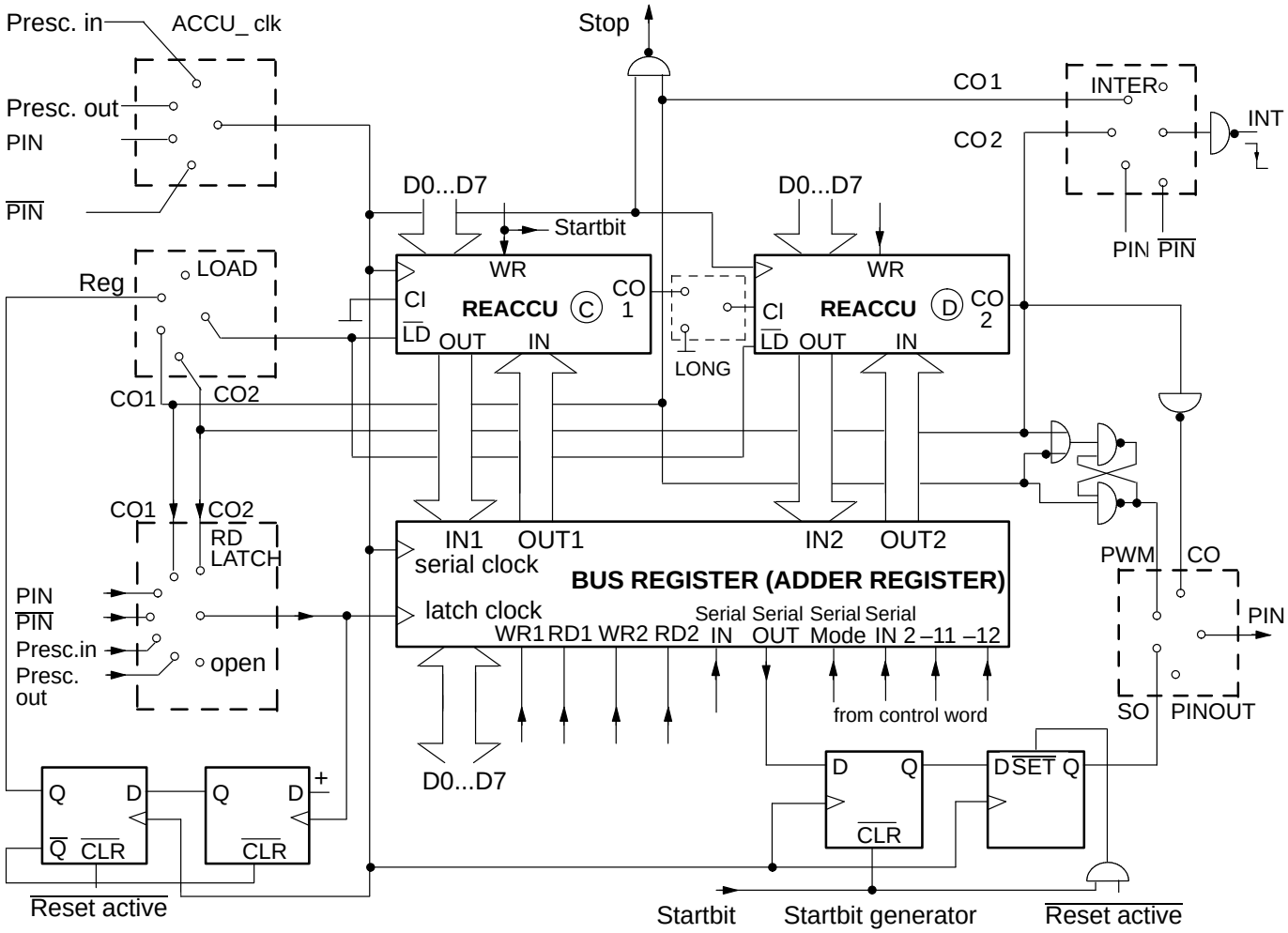


Fig. 2–16: Timer

The three control registers control the internal switches of the timer:

Control register 1

bit 7	SERIAL_2	Second Serial Input Level	
bit 6, 5	Clock	00..	PIN
		01..	fosc.
		10..	Φ2
		11..	disable clock
bit 4	Stop	0...	stop disabled
		1...	carry out accu
bit 3, 2	Start	00..	always active
		01..	edge
		10..	PIN
		11..	$\overline{\text{PIN}}$
bit 1	1/2 load	0...	disabled
		1...	active
bit 0	Edge detector	0...	rising
		1...	falling

Control register 2

bit 7, 6	PIN_out	00..	open
		01..	serial out
		10..	PWM out
		11..	carry accu D
bit 5, 4, 3	Read_Latch	000.	open
		001.	carry accu C
		010.	carry accu D
		011.	PIN
		100.	$\overline{\text{PIN}}$
		101.	prescaler output
		110.	prescaler input
bit 2, 1	accu clock	00..	prescaler input
		01..	prescaler output
		10..	PIN
		11..	$\overline{\text{PIN}}$
bit 0	long	0...	open
		1...	carry out accu C carry in accu D

Control register 3

bit 7, 6, 5	INTER	000.	open
		001.	PIN
		010	$\overline{\text{PIN}}$
		011.	carry accu C
		100.	carry accu D
bit 4, 3	LOAD	00..	open
		01..	carry accu C
		10..	carry accu D
		11..	Reg
bit 2	–11	0...	ACCU C input = bus reg.
		1...	ACCU C input = –1
bit 1	–12	0...	ACCU D input = bus reg.
		1...	ACCU D input = –1
bit 0	Serial mode	0...	mode off
		1...	mode on

2.14. Watchdog

– not active after Reset

– activated when written, cannot be stopped via software

– to retrigger, the watchdog period negated bit by bit must be rewritten within the preset space of time (first write event is also counted)

– triggers reset, the software can identify if Reset was generated by watchdog

– 16 ms to 4 s time-out for 4 MHz system clock

This counter circuit offers hardware support for software problems. It is disabled after reset and enabled with the first write of the desired time value into its register. The value to program is calculated by

$$n = T_{WD} * f_{system} / 65536 - 1$$

with n = watchdog counter value to be programmed for
T_{WD} = the desired watchdog time and
f_{system} = system frequency.

Remarks:
a) To prevent the generation of a 'RESET' by the watchdog before it could be retriggered by the software, *watchdog counter values less than 2 should not be programmed*.
b) The system clock as input of the watchdog counter is *influenced by the system clock prescaler,* determining the CPU speed (register addr. 200 H).

Software can't stop this counter but has to retrigger it by writing the inverted value (one's complement) of the preceding written pattern into its register, which makes unwanted retrigger loops of disturbed software unlikely. These writes have to occur within the time frame (8 ms to 2 s at 8 MHz system clock), defined with the first write.

If no write with the expected pattern occurs within the programmed time period, the watchdog circuit resets the CCU at the end of the time period. There will also be a watchdog reset if another pattern is written instead of the expected one. The software can detect if a reset was generated by the watchdog: Bit 0 of the watchdog register is '0' if the last reset was generated by the watchdog. This bit is reset only with an external reset, e.g. generated by power-on.

Examples:
To set a cycle time of 1 second with 8 MHz system clock the value is 121. This value is calculated as follows:

system frequency: 8 MHz
watchdog cycle time:
65536 / 8 MHz = 8.192 ms,
counter value:
1 s / 8.192 ms = 122.07.

The nearest integer value is 122. Because a 0 loaded into the counter divides by 1, already, the watchdog counter has to be programmed with **122-1 = 121.** With the formula above

$$n = 121 = 1s * 8\ MHz/65536 - 1$$

The software sequences in Assembler could look like this:

Definitions:

```
;constants:
WATCHDOG_TIME           EQU         121
;CCU I/O-address:
watchdog_address        EQU         202H
;variable:
watchdog_value          EQU         30H  ;(address
                           ; of free RAM location)
```

Example 1:

During initialization the watchdog is filled with the desired time-value:

```
 LDA  #WATCHDOG_TIME
 STA  watchdog_address
 STA  watchdog_value    ;memorize
                         ; watchdog pattern
```

In the main loop of the program the watchdog has to be retriggered cyclically:

```
 LDA  watchdog_value
 EOR  #FFH              ;invert bits
 STA  watchdog_address
 STA  watchdog_value    ;memorize new
                         ; watchdog pattern
```

Example 2:

If an interrupt function occurs cyclically, one value may be programmed in the interrupt service routine, while the other is written in the main loop. So both the continuity of executing the interrupt service and the main loop are checked:

During initialization the watchdog shadow variable is filled with the desired time-value:

```
 LDA  #WATCHDOG_TIME
 STA  watchdog_value    ;memorize
                         ; watchdog pattern
```

Sequence in the interrupt function:

```
 LDA  watchdog_value
 CMP  #WATCHDOG_TIME
 BEQ  SKIP_IRQ_WD
;
 STA  watchdog_address
 EOR  #$FF
 STA  watchdog_value
SKIP_IRQ_WD
      . . .
```

Sequence in the main loop:

```
 LDA  watchdog_value
 CMP  #WATCHDOG_TIME
 BNE  SKIP_WD
;
 STA watchdog_address
 EOR #$FF
 STA watchdog_value
SKIP_WD
      ...
```

Remark:
It is important to program the watchdog register with the new value before this value is memorized in the shadow variable, because this procedure could be interrupted by the interrupt, which will program the watchdog with the complementary value.

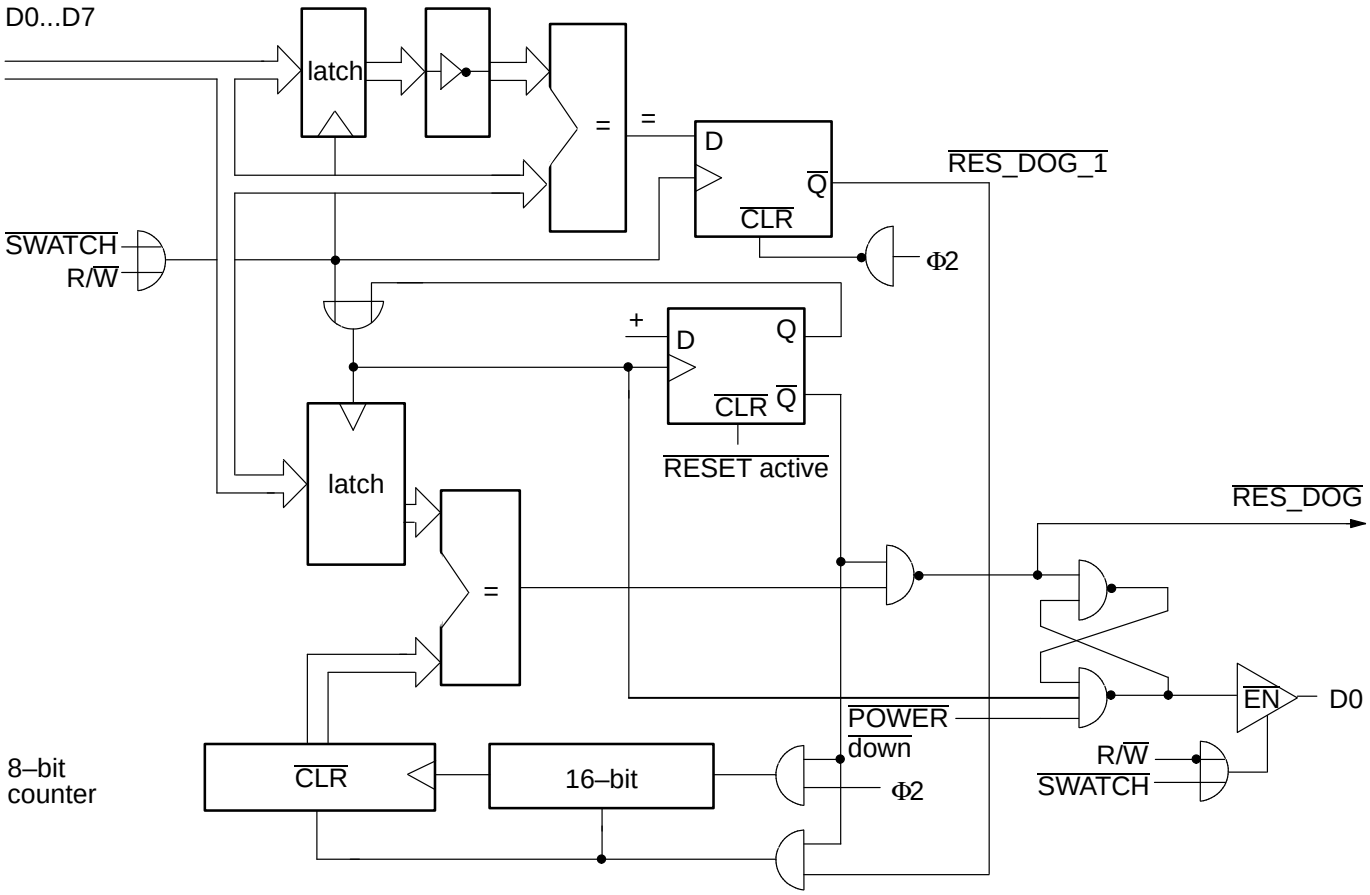


Fig. 2–17: Watchdog

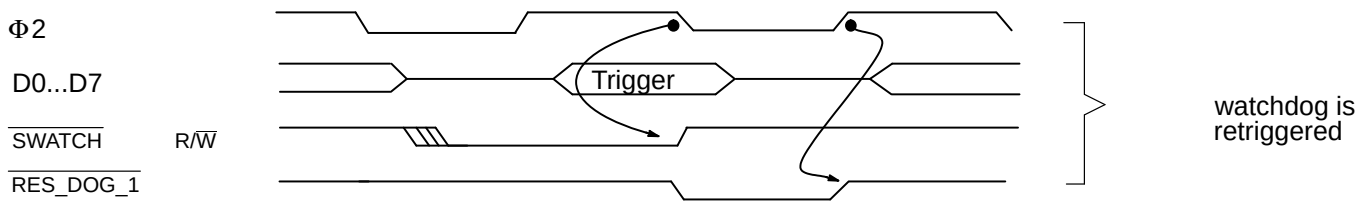


Fig. 2–18: Timing watchdog

2.15. IR-Input

The IR-interface consists of two parallel edge detectors which trigger the rising and falling edge. The respective state of the rising edge triggered flip-flop can be read from D0 (triggered positively), or D1 (triggered negatively). Any read event via the CPU deletes both flip-flops. D2 reflects the status of the IR pin, D3 to D7 are set to 0.

If the CPU is switched off, the IR-Interface is no longer available, as the IR pin is used as output for the interrupt controller. For use as an emulator this function has to be rebuilt externally. The I/O-address designed for the IR-INPUT is treated as an external address when the CPU is switched off, so that the software can remain untouched.

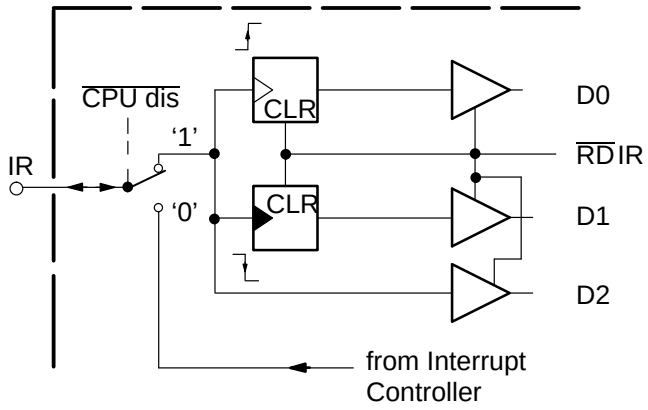


Fig. 2–19: IR input

2.16. Mask Options

There are two mask options:

OSC option: if this option is set, X_1 and X_2 can be used as clock input and output or as XTAL pins, (depending on control word bit 5)

RES option: if this option is set, the reset sources Power on and Clock Supervision are disabled with bit 0 of the test register 2FFH. Default = enabled).

In the production version none of the options is set, in the EMU version both are set.

3. Definitions

3.1. Interrupt Definitions

Interrupt	Source	Vector (low, high byte)
0	TIMER1	FFF6, FFF7
1	TIMER2	FFF4, FFF5
2	TIMER3	FFF2, FFF3
3	IM-BUS1, Master	FFF0, FFF1
4	IM-BUS1, Slave	FFEE, FFEF
5	IM-BUS2, Master	FFEC, FFED
6	IM-BUS2, Slave	FFEA, FFEB
7	P87	FFE8, FFE9
RESET		FFFC, FFFD

3.2. Memory Mappings

RAM	0000H to 01FFH	Page 0, 1
	0300H to 063FH	Page 3, 4, 5, 6
ROM	8000H to FFFFH	(CCU3001 only)
Control byte	FFF9	
I/O	0200 to 02FF	

3.3. I/O Definitions

Address	Function
200H	Clock frequency
201H	Control register
202H	Watchdog
203H	Port 1 Data
204H	Direction Register Port 1
205H	Port 2 Data
206H	Direction Register Port 2
207H	Port 3 Data
208H	Direction Register Port 3
209H	Port 4 Data
20AH	Port 5 Mode Register
20BH	Port 5 Direction Register
20CH	Port 5 Data
20DH	IR-Input
20FH	Port 7 Mode Register
210H	IM-Bus 1 control and status
211H	IM-Bus 1 data transfer rate
213H	IM-Bus 1 master address
214H	IM-Bus 1 master data low byte
215H	IM-Bus 1 master data high byte
216H	IM-Bus 1 slave 1, IM address 02
218H	IM-Bus 1 slave 2, IM-Bus address 03
21AH	IM-Bus 1 slave 3, IM-Bus address 04
21CH	Interrupt controller control byte
21DH	Interrupt controller return byte
21EH	Interrupt controller priorities source 0 & 1
21FH	Interrupt controller priorities source 2 & 3
220H	Interrupt controller priorities source 4 & 5
221H	Interrupt controller priorities source 6 & 7
222H	Timer 1 control byte 1
223H	Timer 1 control byte 2
224H	Timer 1 control byte 3
225H	Timer 1 prescaler low byte
226H	Timer 1 prescaler high byte
228H	Timer 1 accu low byte
229H	Timer 1 accu high byte
22AH	Timer 1 adder low byte
22BH	Timer 1 adder high byte
22CH	Timer 2 control byte 1
22DH	Timer 2 control byte 2
22EH	Timer 2 control byte 3
22FH	Timer 2 prescaler low byte
230H	Timer 2 prescaler high byte
232H	Timer 2 accu low byte
233H	Timer 2 accu high byte
234H	Timer 2 adder low byte
235H	Timer 2 adder high byte
236H	Timer 3 control byte 1
237H	Timer 3 control byte 2
238H	Timer 3 control byte 3
239H	Timer 3 prescaler low byte
23AH	Timer 3 prescaler high byte
23CH	Timer 3 accu low byte
23DH	Timer 3 accu high byte
23EH	Timer 3 adder low byte
23FH	Timer 3 adder high byte
240H	Port 6 Data
241H	Direction Register Port 6
242H	Port 7 Data
243H	Direction Register Port 7
244H	Port 8 Data
245H	Direction Register Port 8
246H	IM-Bus 2 control & status
247H	IM-Bus 2 transfer rate
249H	IM-Bus 2 master address
24AH	IM-Bus 2 master data low
24BH	IM-bus 2 master data high
24CH	IM-bus 2 slave 1, IM address 02
24EH	IM-bus 2 slave 2, IM address 03
250H	IM-bus 2 slave 3, IM address 04
2E0H to 2E7H	External addresses, used for EMU boards
2FEH	Reserved, do not use
2FFH	Reserved for testing purposes

4. Specifications

4.1. Outline Dimensions

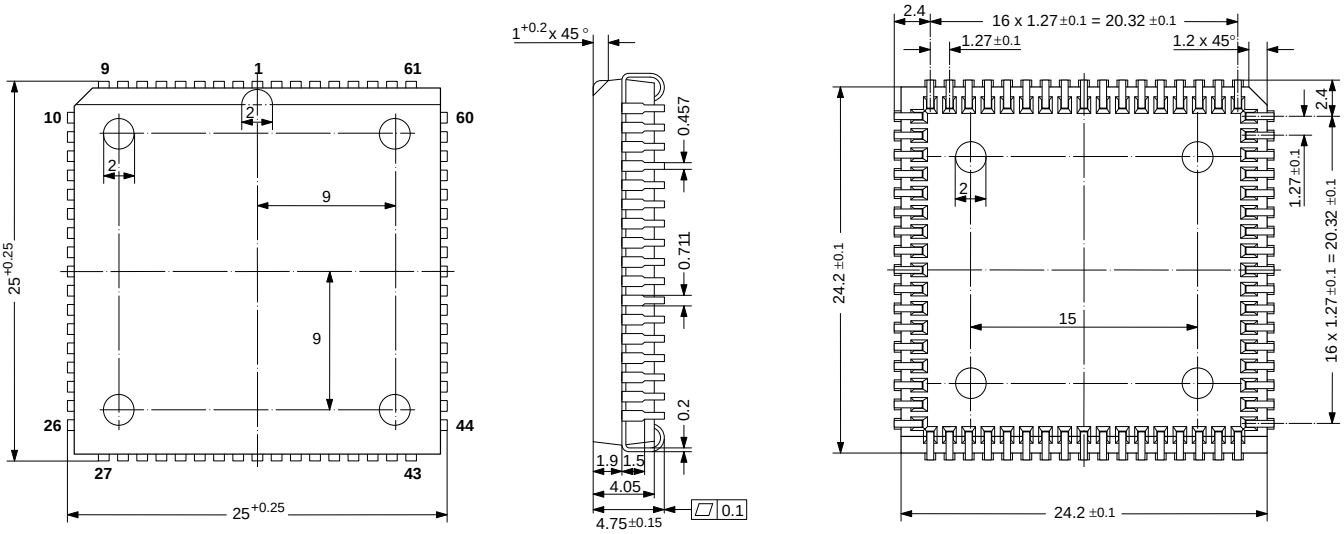


Fig. 4–1: 68-Pin Plastic Leaded Chip Carrier Package **(PLCC68)**
Weight approximately 4.8 g Dimensions in mm

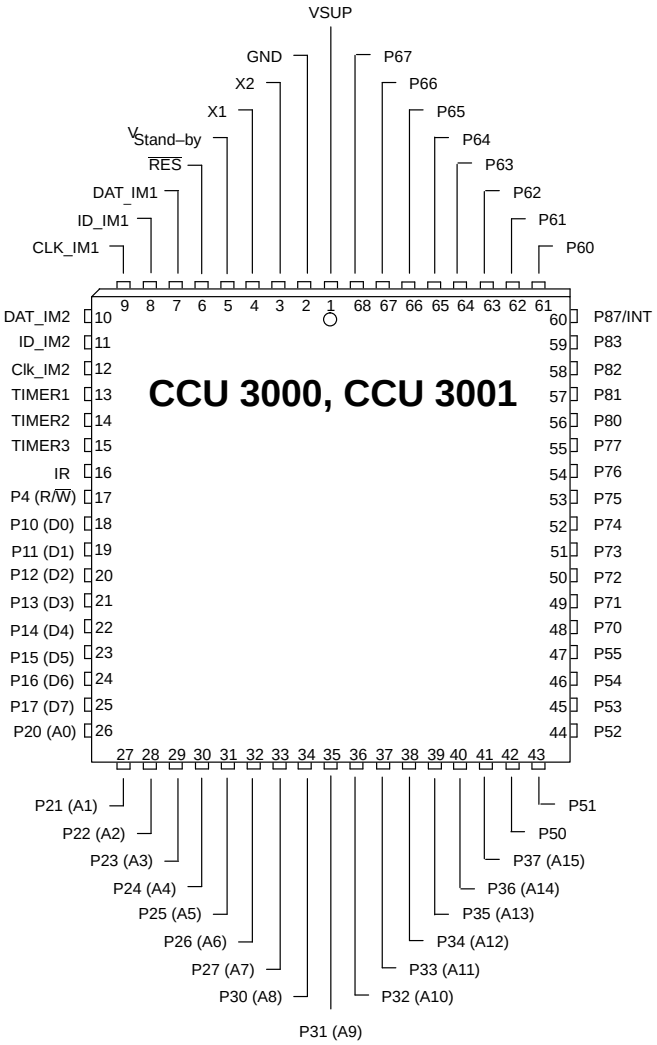


Fig. 4–2: Pinning of the CCU 3000, CCU 3001 in PLCC68 package

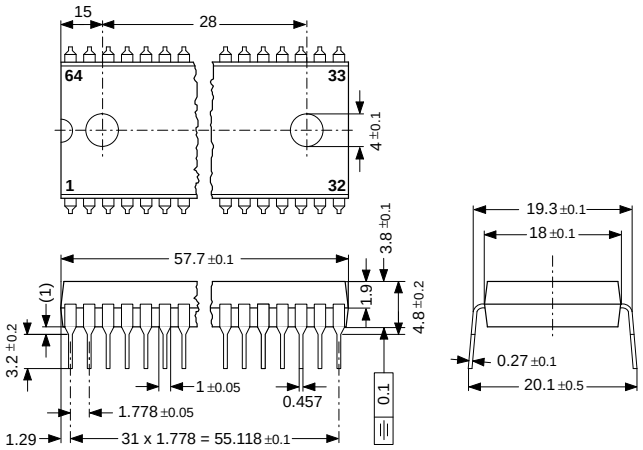


Fig. 4–3:
64-Pin Plastic Shrink Dual Inline Package
(PSDIP64)[1)]
Weight approximately 9.0 g
Dimensions in mm

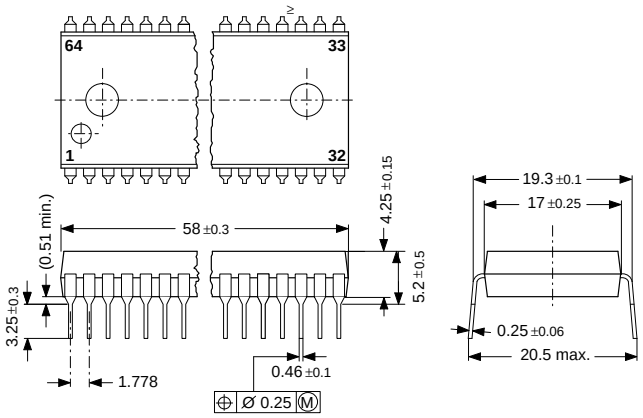


Fig. 4–4:
64-Pin Plastic Shrink Dual Inline Package
(PSDIP64F)[2)]
Weight approximately 9.0 g
Dimensions in mm

[1)] PSDIP64 = Manufactured in Freiburg
[2)] PSDIP64F = Second Source

4.2. Pin Configuration

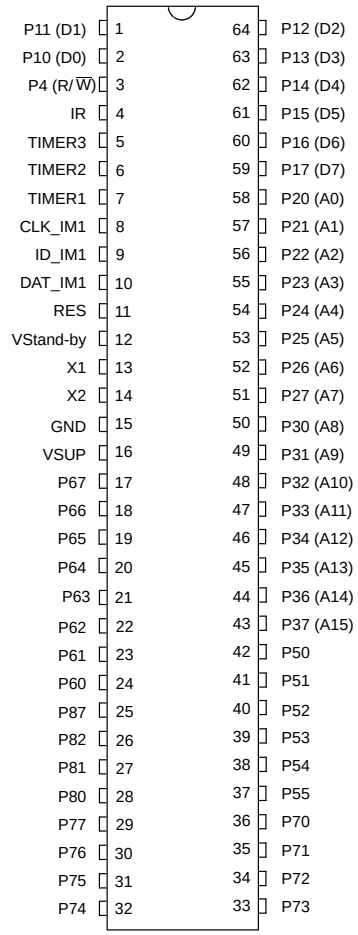


Fig. 4–5: Pinning of the CCU 3000, CCU 3001 in PSDIP64 and PSDIP64F package

4.3. Pin Connections and Short Descriptions

DA = IM bus data line of external devices
ID = IM bus ident line of external devices
CL = IM bus clock line of external devices
X = obligatory; connections depend on application

Pin No.		Con-nection	I: Input O: Output	Pin Name	Short Description
68–pin PLCC	64–pin SDIP				
1	16	+5V	I	V_{SUP}	Supply Voltage
2	15	GND	I	GND	Ground
3	14	Crystal	I/O	X2	Crystal connector 2
4	13	Crystal	I	X1	Crystal connector 1
5	12	+3V to +5V	I	$V_{Stand\text{-}by}$	Standby Supply Voltage
6	11	X	I/O	$\overline{RES}$	Reset input/Reset output
7	10	DA	I/O	DAT_IM1	IM bus 1 data signal
8	9	ID	O	ID_IM1	IM bus 1 ident signal output
9	8	CL	O	CLK_IM1	IM bus 1 clock signal output
10	–	DA	I/O	DAT_IM2	IM bus 2 data signal
11	–	ID	O	ID_IM2	IM bus 2 ident signal output
12	–	CL	O	CLK_IM2	IM bus 2 clock signal output
13	7	X	I/O	TIMER1	Timer 1 signal
14	6	X	I/O	TIMER2	Timer 2 signal
15	5	X	I/O	TIMER3	Timer 3 signal
16	4	external infrared receiver	I	IR	Infrared signal input
17	3	X	I/O (O)	P40 (R/$\overline{W}$)	Port 4 bit 0 (CPU read/$\overline{write}$)
18	2	X	I/O (I/O)	P10 (D0)	Port 1 bit 0 (CPU data bus bit 0)
19	1	X	I/O (I/O)	P11 (D1)	Port 1 bit 1 (CPU data bus bit 1)
20	64	X	I/O (I/O)	P12 (D2)	Port 1 bit 2 (CPU data bus bit 2)
21	63	X	I/O (I/O)	P13 (D3)	Port 1 bit 3 (CPU data bus bit 3)
22	62	X	I/O (I/O)	P14 (D4)	Port 1 bit 4 (CPU data bus bit 4)
23	61	X	I/O (I/O)	P15 (D5)	Port 1 bit 5 (CPU data bus bit 5)

Pin No. 68–pin PLCC	Pin No. 64–pin SDIP	Con-nection	I: Input O: Output	Pin Name	Short Description
24	60	X	I/O (I/O)	P16 (D6)	Port 1 bit 6 (CPU data bus bit 6)
25	59	X	I/O (I/O)	P17 (D7)	Port 1 bit 7 (CPU data bus bit 7)
26	58	X	I/O (O)	P20 (A0)	Port 2 bit 0 (CPU address bus bit 0)
27	57	X	I/O (O)	P21 (A1)	Port 2 bit 1 (CPU address bus bit 1)
28	56	X	I/O (O)	P22 (A2)	Port 2 bit 2 (CPU address bus bit 2)
29	55	X	I/O (O)	P23 (A3)	Port 2 bit 3 (CPU address bus bit 3)
30	54	X	I/O (O)	P24 (A4)	Port 2 bit 4 (CPU address bus bit 4)
31	53	X	I/O (O)	P25 (A5)	Port 2 bit 5 (CPU address bus bit 5)
32	52	X	I/O (O)	P26 (A6)	Port 2 bit 6 (CPU address bus bit 6)
33	51	X	I/O (O)	P27 (A7)	Port 2 bit 7 (CPU address bus bit 7)
34	50	X	I/O (O)	P30 (A8)	Port 3 bit 0 (CPU address bus bit 8)
35	49	X	I/O (O)	P31 (A9)	Port 3 bit 1 (CPU address bus bit 9)
36	48	X	I/O (O)	P32 (A10)	Port 3 bit 2 (CPU address bus bit 10)
37	47	X	I/O (O)	P33 (A11)	Port 3 bit 3 (CPU address bus bit 11)
38	46	X	I/O (O)	P34 (A12)	Port 3 bit 4 (CPU address bus bit 12)
39	45	X	I/O (O)	P35 (A13)	Port 3 bit 5 (CPU address bus bit 13)
40	44	X	I/O (O)	P36 (A14)	Port 3 bit 6 (CPU address bus bit 14)
41	43	X	I/O (O)	P37 (A15)	Port 3 bit 7 (CPU address bus bit 15)
42	42	X	I/O (O)	P50 ($\overline{RD}$ Port 1)	Port 5 bit 0 (CCU read Port 1)
43	41	X	I/O (O)	P51 ($\overline{WR}$ Port 1)	Port 5 bit 1 (CCU write Port 1)
44	40	X	I/O (O)	P52 ($\overline{RD}$ Port 2)	Port 5 bit 2 (CCU read Port 2)
45	39	X	I/O (O)	P53 ($\overline{WR}$ Port 2)	Port 5 bit 3 (CCU write Port 2)
46	38	X	I/O (O)	P54 ($\overline{RD}$ Port 3)	Port 5 bit 4 (CCU read Port 3)
47	37	X	I/O (O)	P55 ($\overline{WR}$ Port 3)	Port 5 bit 5 (CCU write Port 3)
48	36	X	I/O (O)	P70 (Memory Bank Address 0)	Port 7 bit 0 (Memory Bank Address 0)
49	35	X	I/O (O)	P71 (Memory Bank Address 1)	Port 7 bit 1 (Memory Bank Address 1)
50	34	X	I/O (O)	P72 (Memory Bank Address 2)	Port 7 bit 2 (Memory Bank Address 2)
51	33	X	I/O (O)	P73 (Memory Bank Address 3)	Port 7 bit 3 (Memory Bank Address 3)

Pin No. 68–pin PLCC	Pin No. 64–pin SDIP	Con-nection	I: Input O: Output	Pin Name	Short Description
52	32	X	I/O (O)	P74 (Memory Bank Address 4)	Port 7 bit 4 (Memory Bank Address 4)
53	31	X	I/O (O)	P75 (Memory Bank Address 5)	Port 7 bit 5 (Memory Bank Address 5)
54	30	X	I/O (O)	P76 ($\overline{R}$/W)	Port 7 bit 6 (CPU read/write)
55	29	X	I/O (O)	P77 (Power-Down Control)	Port 7 bit 7 (Power-Down Control)
56	28	X	I/O	P80	Port 8 bit 0
57	27	X	I/O	P81	Port 8 bit 1
58	26	X	I/O	P82	Port 8 bit 2
59	–	X	I/O	P83	Port 8 bit 3
60	25	X	I/O /I	P87/INT	Port 8 bit 7 /interrupt input
61	24	X	I/O	P60	Port 6 bit 0
62	23	X	I/O	P61	Port 6 bit 1
63	22	X	I/O	P62	Port 6 bit 2
64	21	X	I/O	P63	Port 6 bit 3
65	20	X	I/O	P64	Port 6 bit 4
66	19	X	I/O	P65	Port 6 bit 5
67	18	X	I/O	P66	Port 6 bit 6
68	17	X	I/O	P67	Port 6 bit 7

4.4. Pin Descriptions

CCU 3000, CCU 3001 Pin Descriptions.
Pin numbers refer to the 68–pin PLCC housing.

The functions of some pins are influenced by bit 4 of the CCU control register (addr. 201H, copied from FFF9H at reset: CCU control register bit 4 = '1' switches the CCU in *Port Mode* , CCU control register bit 4 = '0' switches the CCU in *Bus Mode*.

In addition, some port bit functions may be changed between *Normal Mode* and *Special Mode* by setting the specific bit in its port mode registers.

Pin 1: V_{sup}: +5V power supply

Pin 2: GND: Digital ground

Pin 3: X2: Second Crystal connector

Pin 4: X1: First Crystal connector

Pin 5: $V_{Stand–by:}$ +5V Stand–by Supply Voltage

Pin 6: RES\: CCU Reset input / output (open drain)

Pin 7: DAT_IM1: IM bus 1 data signal (I/O)

Pin 8: ID_IM1: IM bus 1 ident signal output

Pin 9: CLK_IM1: IM bus 1 clock signal output

Pin 10: DAT_IM2: IM bus 2 data signal (I/O)

Pin 11: ID_IM2: IM bus 2 ident signal output

Pin 12: CLK_IM2: IM bus 2 clock signal output

Pin 13: TIMER1: Timer 1 signal (I/O)

Pin 14: TIMER2: Timer 2 signal (I/O)

Pin 15: TIMER3: Timer 3 signal (I/O)

Pin 16: IR: Infrared signal input

Pin 17: P40 or R/W\:
in *Port Mode:* Port 4 Bit 0
in *Bus Mode*: CPU read/not write output

Pin 18 : P10 or data bit 0:
in *Port Mode:* Port 1 Bit 0
in *Bus Mode*: CPU data bit 0

Pin 19 : P11 or data bit 1:
in *Port Mode:* Port 1 Bit 1
in *Bus Mode*: CPU data bit 1

Pin 20 : P12 or data bit 2:
in *Port Mode:* Port 1 Bit 2
in *Bus Mode*: CPU data bit 2

Pin 21 : P13 or data bit 3:
in *Port Mode:* Port 1 Bit 3
in *Bus Mode*: CPU data bit 3

Pin 22 : P14 or data bit 4:
in *Port Mode:* Port 1 Bit 4
in *Bus Mode*: CPU data bit 4

Pin 23 : P15 or data bit 5:
in *Port Mode:* Port 1 Bit 5
in *Bus Mode*: CPU data bit 5

Pin 24 : P16 or data bit 6:
in *Port Mode:* Port 1 Bit 6
in *Bus Mode*: CPU data bit 6

Pin 25 : P17 or data bit 7:
in *Port Mode:* Port 1 Bit 7
in *Bus Mode*: CPU data bit 7

Pin 26 : P20 or address bit 0:
in *Port Mode:* Port 2 Bit 0
in *Bus Mode*: CPU address bit 0

Pin 27 : P21 or address bit 1:
in *Port Mode:* Port 2 Bit 1
in *Bus Mode*: CPU address bit 1

Pin 28 : P22 or address bit 2:
in *Port Mode:* Port 2 Bit 2
in *Bus Mode*: CPU address bit 2

Pin 29 : P23 or address bit 3:
in *Port Mode:* Port 2 Bit 3
in *Bus Mode*: CPU address bit 3

Pin 30 : P24 or address bit 4:
in *Port Mode:* Port 2 Bit 4
in *Bus Mode*: CPU address bit 4

Pin 31 : P25 or address bit 5:
in *Port Mode:* Port 2 Bit 5
in *Bus Mode*: CPU address bit 5

Pin 32 : P26 or address bit 6:
in *Port Mode:* Port 2 Bit 6
in *Bus Mode*: CPU address bit 6

Pin 33 : P27 or address bit 7:
in *Port Mode:* Port 2 Bit 7
in *Bus Mode*: CPU address bit 7

Pin 34 : P30 or address bit 8:
in *Port Mode:* Port 3 Bit 0
in *Bus Mode*: CPU address bit 8

Pin 35 : P31 or address bit 9:
in *Port Mode:* Port 3 Bit 1
in *Bus Mode*: CPU address bit 9

Pin 36 : P32 or address bit 10:
in *Port Mode:* Port 3 Bit 2
in *Bus Mode*: CPU address bit 10

Pin 37 : P33 or address bit 11:
in *Port Mode:* Port 3 Bit 3
in *Bus Mode*: CPU address bit 11

Pin 38 : P34 or address bit 12:
in *Port Mode:* Port 3 Bit 4
in *Bus Mode*: CPU address bit 12

Pin 39 : P35 or address bit 13:
in *Port Mode:* Port 3 Bit 5
in *Bus Mode*: CPU address bit 13

Pin 40 : P36 or address bit 14:
in *Port Mode:* Port 3 Bit 6
in *Bus Mode*: CPU address bit 14

Pin 41 : P37 or address bit 15:
in *Port Mode:* Port 3 Bit 7
in *Bus Mode*: CPU address bit 15

Pin 42 : P50 or $\overline{\text{RD}}$Port1\:
in *Port Mode:*
in *Normal Mode*: Port 5 Bit 0 (open drain output)
in *Special Mode*: read port 1 (low active)
in *Bus Mode*:
in *Normal Mode*: Port 5 Bit 0 (open drain output)
in *Special Mode*: read port 1 (low active)

Pin 43 : P51 or $\overline{\text{WR}}$Port1\:
in *Port Mode:*
in *Normal Mode*: Port 5 Bit 1 (open drain output)
in *Special Mode*: write port 1 (low active)
in *Bus Mode*:
in *Normal Mode*: Port 5 Bit 1 (open drain output)
in *Special Mode*: write port 1 (low active)

Pin 44 : P52 or $\overline{\text{RD}}$Port2\:
in *Port Mode:*
in *Normal Mode*: Port 5 Bit 2 (open drain output)
in *Special Mode*: read port 2 (low active)
in *Bus Mode*:
in *Normal Mode*: Port 5 Bit 2 (open drain output)
in *Special Mode*: read port 2 (low active)

Pin 45 : P53 or $\overline{\text{WR}}$Port2\:
in *Port Mode:*
in *Normal Mode*: Port 5 Bit 3 (open drain output)
in *Special Mode*: write port 2 (low active)
in *Bus Mode*:
in *Normal Mode*: Port 5 Bit 3 (open drain output)
in *Special Mode*: write port 2 (low active)

Pin 46 : P54 or $\overline{\text{RD}}$Port3\:
in *Port Mode:*
in *Normal Mode*: Port 5 Bit 4 (open drain output)
in *Special Mode*: read port 3 (low active)
in *Bus Mode*:
in *Normal Mode*: Port 5 Bit 4 (open drain output)
in *Special Mode*: read port 3 (low active)

Pin 47 : P55 or $\overline{\text{WR}}$Port3\:
in *Port Mode:*
in *Normal Mode*: Port 5 Bit 5 (open drain output)
in *Special Mode*: write port 3 (low active)
in *Bus Mode*:
in *Normal Mode*: Port 5 Bit 5 (open drain output)
in *Special Mode*: write port 3 (low active)

Pin 48 : P70 or Memory Bank Address 0:
in *Port Mode:*
in *Normal Mode*: Port 7 Bit 0
in *Special Mode*: Memory Bank Address 0
in *Bus Mode*:
in *Normal Mode*: Port 7 Bit 0
in *Special Mode*: Memory Bank Address 0

Pin 49 : P71 or Memory Bank Address 1:
in *Port Mode:*
in *Normal Mode*: Port 7 Bit 1
in *Special Mode*: Memory Bank Address 1
in *Bus Mode*:
in *Normal Mode*: Port 7 Bit 1
in *Special Mode*: Memory Bank Address 1

Pin 50 : P72 or Memory Bank Address 2:
in *Port Mode:*
in *Normal Mode*: Port 7 Bit 2
in *Special Mode*: Memory Bank Address 2
in *Bus Mode*:
in *Normal Mode*: Port 7 Bit 2
in *Special Mode*: Memory Bank Address 2

Pin 51 : P73 or Memory Bank Address 3:
in *Port Mode:*
in *Normal Mode*: Port 7 Bit 3
in *Special Mode*: Memory Bank Address 3
in *Bus Mode*:
in *Normal Mode*: Port 7 Bit 3
in *Special Mode*: Memory Bank Address 3

Pin 52 : P74 or Memory Bank Address 4:
in *Port Mode:*
in *Normal Mode*: Port 7 Bit 4
in *Special Mode*: Memory Bank Address 4
in *Bus Mode*:
in *Normal Mode*: Port 7 Bit 4
in *Special Mode*: Memory Bank Address 4

Pin 53 : P75 or Memory Bank Address 5:
in *Port Mode:*
in *Normal Mode*: Port 7 Bit 5
in *Special Mode*: Memory Bank Address 5
in *Bus Mode*:
in *Normal Mode*: Port 7 Bit 5
in *Special Mode*: Memory Bank Address 5

Pin 54 : P76 or inverted CPU R/W\:
in *Port Mode:*
in *Normal Mode*: Port 7 Bit 6
in *Special Mode*: inverted CPU R/W\,
i.e.: low active at read
in *Bus Mode*:
in *Normal Mode*: Port 7 Bit 6
in *Special Mode*: inverted CPU R/W\,
i.e.: low active at read

Pin 55 : P77 or Power-down Control:
in *Port Mode:*
in *Normal Mode*: Port 7 Bit 7
in *Special Mode*: Power-down Control External Memory (high active)
in *Bus Mode*:
in *Normal Mode*: Port 7 Bit 7
in *Special Mode*: Power-down Control External Memory (high active)

Pin 56 : P80: Port 8 Bit 0

Pin 57 : P81: Port 8 Bit 1

Pin 58 : P82: Port 8 Bit 2

Pin 59 : P83: Port 8 Bit 3

Pin 60 : P87/INT:
Port 8 Bit 7 and interrupt input (interrupt controller source 7)

Pin 61 : P60: Port 6 Bit 0

Pin 62 : P61: Port 6 Bit 1

Pin 63 : P62: Port 6 Bit 2

Pin 64 : P63: Port 6 Bit 3

Pin 65 : P64: Port 6 Bit 4

Pin 66 : P65: Port 6 Bit 5

Pin 67 : P66: Port 6 Bit 6

Pin 68 : P67: Port 6 Bit 7

4.5. Pin Circuits

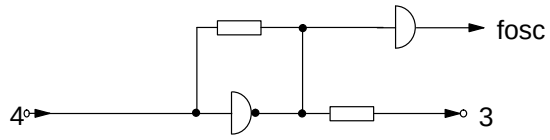


Fig. 4–6: X1, X2

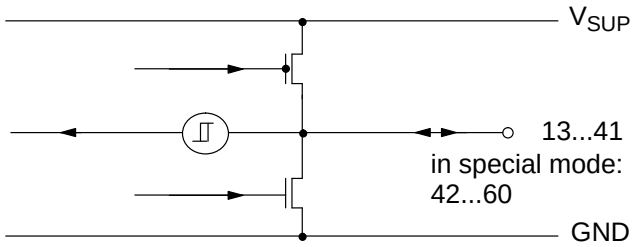


Fig. 4–10: P1, P2, P3, Timer (1, 2, 3), IR (Input only), in special mode: P5, P7, P8

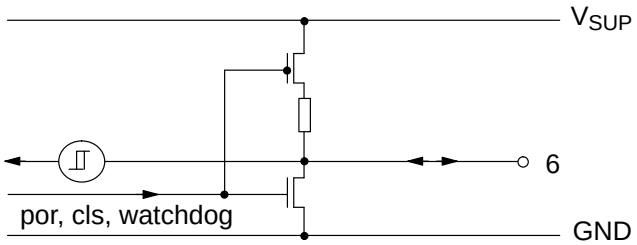


Fig. 4–7: Reset

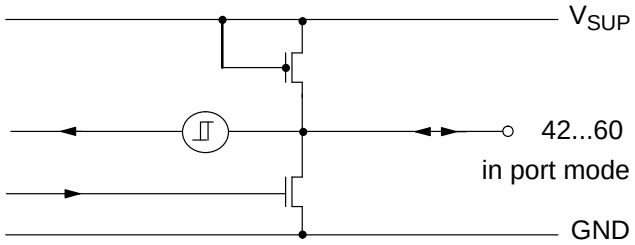


Fig. 4–11: P5, P6, P7, P8 in port mode

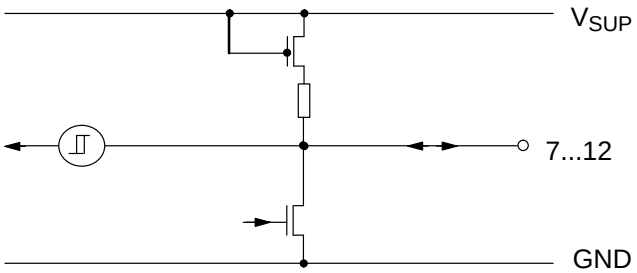


Fig. 4–8: IM bus

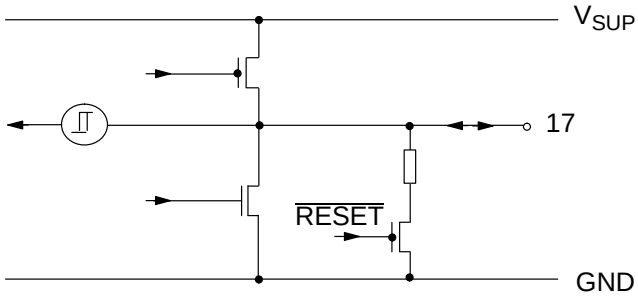


Fig. 4–12: R/$\overline{W}$, / P4, CCU 3001

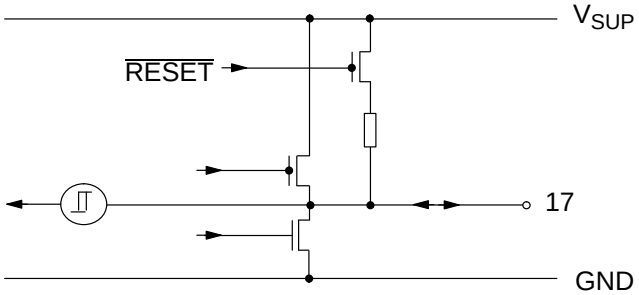


Fig. 4–9: R/$\overline{W}$, CCU 3000

4.6. Electrical Characteristics

All voltages refer to ground.

4.6.1. Absolute Maximum Ratings

Symbol	Parameter	Pin	Min.	Max.	Unit
T_A	Ambient Operating Temperature	–	0	65	°C
T_S	Storage Temperature	–	–40	+125	°C
V_{SUP}	Supply Voltage	1	–0.5	6	V
V_I	Input Voltage	4, 6, 16, 13 to 25, 42 to 68	–0.3 V	V_{SUP}+0.3 V	–
P_{max}	Maximum Power Dissipation	–	–	500	mW

The total sum of all the sink currents of all ports together must not exceed 80 mA I_{outlow} and 280 mA – I_{out} high at any time. Stresses beyond those listed in the “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only. Functional operation of the device at these or any other conditions beyond those indicated in the “Recommended Operating Conditions/Characteristics” of this specification is not implied. Exposure to absolute maximum ratings conditions for extended periods may affect device reliability.

4.6.2. Recommended Operating Conditions at T_A = 0 °C to 65°C

Symbol	Parameter	Pin	Min.	Typ.	Max.	Unit
V_{SUP}	Supply Voltage	1	4.75	–	5.25	V
f_{CLK}	Clock Frequency	3, 4	0.5	–	8	MHz

4.6.3. Recommended Crystal Characteristics at $C_{XTAL1} = C_{XTAL2}$ = 22 pF; $C_{stray} \leq$ 2 pF; $C_L \approx$13 pF

Symbol	Parameter	Min.	Typ.	Max.	Unit
T_A	Ambient Operating Temperature	–20	–	+85	°C
f_p	Parallel Resonance Frequency @ C_L = 13 pF	–	4–8	–	MHz
R_r	Series Resistance	–	–	40 [8MHz] 150 [4 MHz]	Ω
C_0	Shunt Capacitance	–	–	7.0	pF
C_1	Motional Capacitance	–	–	20	fF
P	Rated Drive Level	–	0.02	–	mW
f_p / f_H	Spurious Frequency Attenuation	20	–	–	dB

4.6.4. DC Characteristics at T_A = 0 °C to 65°C, V_{SUP} = 5V, f_{CLK} = 8 MHz

Symbol	Parameter	Pin No.	Min.	Typ.	Max.	Unit	Comment
I_{SUP} CCU 3000, CCU 3000-I	Supply Current (no external load)	1	–	8/16	15/30	mA	@ 4/8 MHz
I_{SUP} CCU 3001 CCU 3001-I	Supply Current (no external load)	1	–	14/28	20/40	mA	@ 4/8 MHz
V_{ILH}	Input Low to High Trigger Level	4,6, 7–12, 13–25, 42–68	0.38* V_{SUP}	2,05	0.56* V_{SUP}	V	Schmitt-Trigger Inputs
V_{IHL}	Input High to Low Trigger Level		0.2* V_{SUP}	1.15	0.29* V_{SUP}	V	Schmitt-Trigger Inputs
V_{IHYST}	Input Hysteresis	4,6, 7–25, 42–68	0.1* V_{SUP}	0.9	0.27* V_{SUP}	V	Schmitt-Trigger Inputs
V_{BOH}	Bus Ports and Timer Output High Voltage	13–15 17–41	2.8	–	–	V	
V_{BOL}	Bus Ports and Timer Output Low Voltage		–	–	0.4	V	
I_{BOH}	Bus Ports and Timer Output High Current	13–151 7–41	–	–	2	mA	
I_{BOL}	Bus Ports and Timer Output Low Current		–	–	5	mA	
I_{P5OL}	P5 Output Low Current	42–47	–	–	5	mA	
I_{P6OL}	P6 Output Low Current	61–68	–	–	25	mA	
I_{P7OL}	P7 Output Low Current	48–55	–	–	5	mA	
I_{P8OL}	P8 Output Low Current	56–60	–	–	5	mA	
I_{OHL}	Output Leakage Current	42–60	–	–	1	µA	
I_{P5P7OH}	Output High Current Special Mode, P5 to P7	42–55	–	–	2	mA	
V_{P5P7OH}	Output High Voltage Special Mode, P5 to P7	42–55	2.8	–	–	V	
V_{OL}	Output Low Voltage at I_{OL} = 5 mA	42–60	–	–	0.4	V	
V_{OL}	Output Low Voltage at I_{OL} = 25 mA	61–68	–	–	0.55	V	
V_{Stby}	Stand-by Voltage RAM	5	3	–	–	V	
I_{Stby}	Stand-by Current RAM	5	–	–	1	µA	@ V_{DD} = 0 V, V_{Stby} = 5V
$V_{Pow\ on}$	Power-on Voltage	1	4.2	–	4.75	V	

Bus Ports: P1, P2, P3, P4.

4.6.5. Using external devices

To avoid collision on the data bus during direction changes, the CCU data bus out buffers (active during PHI2 = '1' only) are disabled before the address, the R/$\overline{W}$ and the $\overline{R}$/W line changes ($t_{DHW} < t_{AH)}$. This guarantees that no collision happens on the bus if the output drives of the external devices (ROM, RAM, Ports) are controlled with the R/$\overline{W}$ or $\overline{R}$/W signal and a read cycle follows a write cycle.

Important: In a write cycle the data-out drivers of the CCU set up the data bus lines. Then they leave these lines so that no drivers are active on the data bus. A few ns later the R/$\overline{W}$ or the P5 select signal latch the data into the external device (Port out or Write into RAM). The same signal is used to enable the output drivers of external devices for reading so that another few ns later they drive the bus. The DATA BUS is used as DATA MEMORY for a few ns. This is the only way to make sure that, independent from the loads on the CCU address, data and control lines collisions are avoided and a maximum of access time is available for the memory.

IF YOU WANT TO WRITE TO EXTERNAL DEVICES THE DATA BUS MUST BE IN THE TRISTATE MODE DURING WRITE OPERATIONS OF THE CCU. No pull-up or pull-down resistors are allowed.

Even in a good layout the capacitive load on the data bus is approx. 20 pF (2* pin capacity and layout). Even in the worst case of a 1 MΩ leakage the time constant is approx. 20 µs. The max. time between disabling the bus drivers and the rising edge of R/$\overline{W}$ is 20 ns.

4.6.6. AC Characteristics at T_A = 0 °C to 65°C, V_{SUP} = 5 V, f_{CLK} = 8 MHz, **Cl = 0 pF**
External Loads: add 0.75 ns/pF for controller output lines

Symbol	Parameter	Pin	Min.	Max.	Unit
t_{cyc}	Cycle Time (Processor)	3	125	2000	ns
t_{PWL}	Pulse Width Low	3	60	1000	ns
t_{AH}	Address Hold Time	26–41	10	22	ns
t_{ADS}	Address Setup Time Read		15	34	ns
t_{DSR}	Data Setup Time Read	18–25	20	–	ns
t_{MDS}	Write Data Delay		10	29	ns
t_{DHW}	Write Data Hold Time		9	16	ns
t_{DHR}	Read Data Hold Time		10	–	ns
t_{RWH}	Read/$\overline{Write}$ Hold Time	17	10	24	ns
t_{WRH}	$\overline{Read}$/Write Hold Time (P76 special mode)	17	13	34	ns
t_{P5S}	Delay P1 to P3 Select Lines on P5	42–47	12	26	ns
$t_{X1\Phi2}$	Delay X1 to internal Φ2	4	7	15	ns
$t_{\Phi2X2}$	Delay Φ2 internal to X2	3	5	10	ns

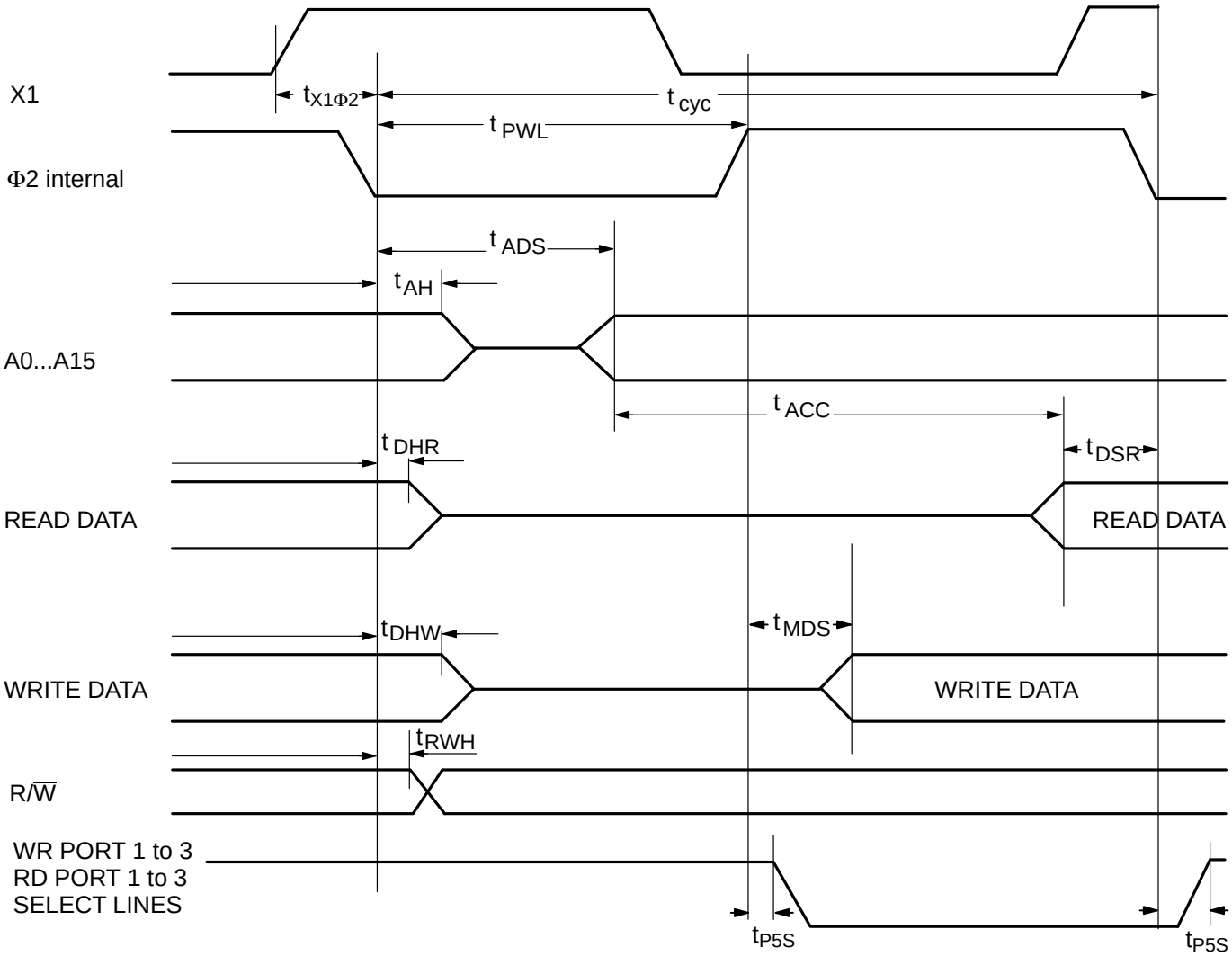


Fig. 4–13: AC timing

4.6.7. IM Bus Waveforms

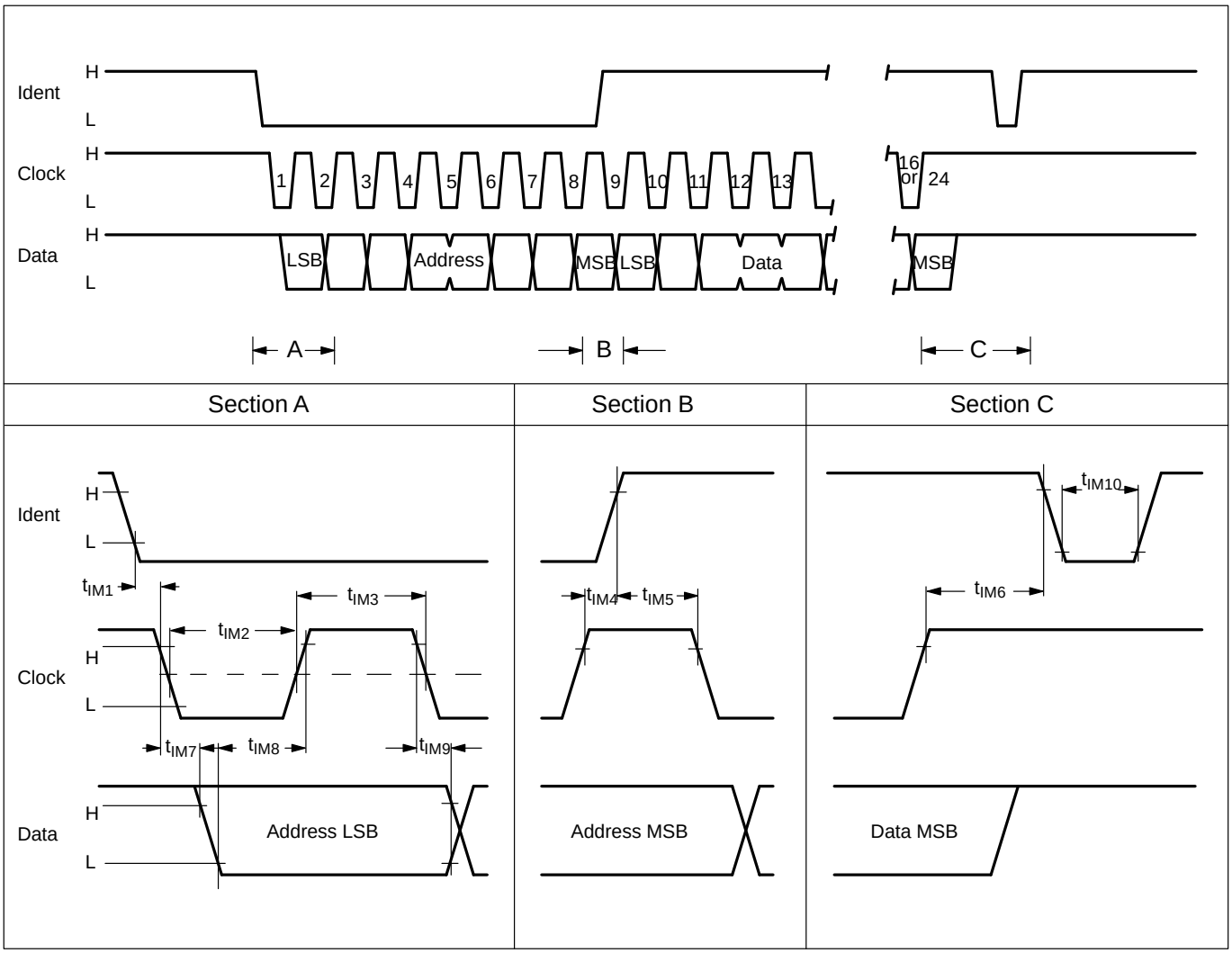


Fig. 4–14: IM bus waveforms

4.6.8. Description of the IM Bus

The INTERMETALL Bus (IM bus for short) was designed to control the DIGIT 2000 ICs by the CCU Central Control Unit. Via this bus the CCU can write data to the ICs or read data from them. This means that the CCU acts as a master, whereas all controlled ICs have purely slave status.

The IM bus consists of three lines for the signals Ident (ID), Clock (DL) and Data (D). The clock frequency range is 50 Hz to 1 MHz. Ident and clock are unidirectional from the CCU to the slave ICs, Data is bidirectional. Bidirectionality is achieved by using open-drain outputs. The 2.5 to 1 kOhm pull-up resistor common to all outputs must be connected externally.

The timing of a complete IM bus transaction is shown in Fig. 4–14. In the non-operative state the signals of all three bus lines are High. To start a transaction the CCU sets the ID signal to Low level, indicating an address transmission, and sets the CL signal to Low level as well as to switch the first bit on the Data line. Then eight address bits are transmitted, beginning with the the LSB. Data takeover in the slave ICs occurs at the positive edge of the clock signal. At the end of the address byte the ID signal switches to High, initiating the address comparison in the slave circuits. In the addressed slave the IM bus interface switches over to Data read or write, because these functions are correlated to the address. Also controlled by the address the CCU now transmits eight or sixteen clock pulses, and accordingly one or two bytes of data are written into the addressed IC or read out from it, beginning with the LSB.

The completion of the bus transaction is signalled by a short Low state pulse of the ID signal. This initiates the storing of the transferred data.

For future software compatibility, the CCU must write a zero into all bits not currently used. When reading undefined or unused bits, the CCU must adopt “don’t care” behavior.

4.6.9. Recommended Operating Conditions of IM Bus

Symbol	Parameter	Pin No.	Min.	Typ.	Max.	Unit
V_{IML}	IM bus Low Voltage	Data 7, 10	–	–	0.8	V
V_{IMH}	IM bus High Voltage		2.4	–	–	V
R_{ext}	External Pull-Up Resistor		1	–	2.5	kΩ
I_{IMOL}	IM bus Output Low Current		–	–	5	mA
$f_{\Phi I}$	ΦI IM bus Clock Frequency	Clock 9, 12	0.05	–	1000	kHz
t_{IM1}	ΦI Clock Delay Time after IM bus Ident		> 0	–	–	–
t_{IM2}	ΦI Clock Low Pulse Time		500	–	–	ns
t_{IM3}	ΦI Clock High Pulse Time		500	–	–	ns
t_{IM4}	ΦI Clock Setup Time before Ident High		> 0	$0.5 \cdot t_{IM3}$	–	ns
t_{IM5}	ΦI Clock Hold Time after Ident High		250	$0.5 \cdot t_{IM3}$	–	ns
t_{IM6}	ΦI Clock Setup Time before Ident End-Pulse		1	$t_{IM2} + t_{IM3}$	–	µs
t_{IM7}	IM bus Data Delay Time after ΦI Clock		> 0	–	–	ns
t_{IM8}	IM bus Data Setup Time before ΦI Clock	Clock, Data 7, 9, 10, 12	> 0	–	–	ns
t_{IM9}	IM bus Data Hold Time after ΦI Clock		> 0	–	–	ns
t_{IM10}	IM bus Ident End-Pulse Low Time	Ident 8, 11	1	$t_{IM2} + t_{IM3}$	–	µs

4.6.10. Registers

0200H	System Clock Prescaler		
Bit	**Reset**	**Read**	**Write**
7 to 3	0	x	Divisor value –1
2	1	x	
1	0	x	
0	0	x	

0201H	Control Register		
Bit	**Reset**	**Read**	**Write**
7	copy from addr. FFF9H	x	no function – set to ‘1’ (to keep compatibility)
6		x	no function – set to ‘1’ (to keep compatibility)
5		x	no function – set to ‘1’ (to keep compatibility)
4		external bus: ‘1’ = bus on ports 0, 1, 2 disabled	Bus disable: ‘1” = disable bus on ports 0, 1, 2
3		R/W signal / Port4: ‘0’ = R/W, ‘1’ = P40	R/W signal / Port4: ‘0’ = R/W, ‘1’ = P40
2		internal ROM: ‘1’ = internal ROM enabled	ROM enable: ‘1’ = enable internal ROM
1		internal RAM: ‘1’ = internal RAM enabled	RAM enable: ‘1’ = enable internal RAM
0		internal CPU: ‘1’ = internal CPU enabled	CPU enable: ‘1’ = enable internal CPU

0202H	Watchdog Control and Status		
Bit	**Reset**	**Read**	**Write**
7	x	x	Watchdog time value = $(\frac{f_{system}}{65536} * Twd) - 1 = n$ $T_{wd} = (n+1) * \frac{65536}{f_{system}}$ (don’t use setting n<2!!) with f_{system} = 4 MHz: $n = n_{min} = 2 \Rightarrow T_{wdmin} = 49.152$ ms $n = n_{max} = 255 \Rightarrow T_{wdmax} = 4.17792$ s min. $\Delta n = 1 \Rightarrow$ min. ΔT_{wd} =16.384 ms
6	x	x	
5	x	x	
4	x	x	
3	x	x	
2	x	x	
1	x	x	
0	1/0	‘0’: last RESET was generated by watchdog	

0203H	Port 1 Data Register		
Bit	**Reset**	**Read**	**Write**
7 to 0	0	Port 1 Data	Port 1 Data

0204H	Port 1 Direction Register		
Bit	**Reset**	**Read**	**Write**
7 to 0	1	x	‘1’ = input, ‘0’ = output mode

0205H	Port 2 Data Register		
Bit	**Reset**	**Read**	**Write**
7 to 0	0	Port 2 Data	Port 2 Data

0206H	Port 2 Direction Register		
Bit	**Reset**	**Read**	**Write**
7 to 0	1	x	‘1’ = input, ‘0’ = output mode

0207H	Port 3 Data Register		
Bit	**Reset**	**Read**	**Write**
7 to 0	0	Port 3 Data	Port 3 Data

0208H	Port 3 Direction Register		
Bit	**Reset**	**Read**	**Write**
7 to 0	1	x	‘1’ = input, ‘0’ = output mode

0209H	Port 4 Data Register		
Bit	**Reset**	**Read**	**Write**
7 to 1	x	x	x
0	0	Port 4 Data (bit 0 only)	Port 4 data (bit 0, only)

020AH	Port 5 Mode		
Bit	**Reset**	**Read**	**Write**
7 to 6	x	x	x
5 to 0	x	Port 4 Data (bit 0 only)	‘1’ = port select mode, ‘0’ = port mode

020BH	Port 5 Direction Register		
Bit	**Reset**	**Read**	**Write**
7 to 6	x	x	x
5 to 0	1	x	‘1’ = input, ‘0’ = output mode

020CH	Port 5 Data Register		
Bit	**Reset**	**Read**	**Write**
7 to 6	x	x	x
5	0	Data bit 5	Data bit 5
4	0	Data bit 4	Data bit 4
3	0	Data bit 3	Data bit 3
2	0	Data bit 2	Data bit 2
1	0	Data bit 1	Data bit 1
0	0	Data bit 0	Data bit 0

020DH	IR Input		
Bit	**Reset**	**Read**	**Write**
7 to 3	x	x	x
2	x	IR–pin logic level	x
1	x	‘1’ if a falling edge occurred since last read, else ‘0’	x
0	x	‘1’ if a rising edge occurred since last read, else ‘0’	x

020FH	Port 7 Mode Register		
Bit	**Reset**	**Read**	**Write**
7	0	x	‘0’ = port mode, ‘1’ = power down control for external memory
6	0	x	‘0’ = port mode, ‘1’ = \R/W output
5	0	x	‘0’ = port mode, ‘1’ = banking addr. bit 5
4	0	x	‘0’ = port mode, ‘1’ = banking addr. bit 4
3	0	x	‘0’ = port mode, ‘1’ = banking addr. bit 3
2	0	x	‘0’ = port mode, ‘1’ = banking addr. bit 2
1	0	x	‘0’ = port mode, ‘1’ = banking addr. bit 1
0	0	x	‘0’ = port mode, ‘1’ = banking addr. bit 0

0210H	IM Bus 1 Control and Status Register			
Bit	**Reset**	**Read**	**Write**	
7 to 4	x	x	x	
3	0	‘1’ = 1 byte received in slave register 3 (IM bus address 4)	‘1’ = read word via IM bus (master)	(bits 0 to 3: ‘0000’= reset IM bus interface)
2	0	‘1’ = 1 byte received in slave register 2 (IM bus address 3)	‘1’ = read byte via IM bus (master)	
1	0	‘1’ = 1 byte received in slave register 1 (IM bus address 2)	‘1’ = write word via IM bus (master)	
0	0	‘1’ = IM bus (master) busy	‘1’ = write byte via IM bus (master)	

0211H	IM Bus 1 Data Transfer Rate Register		
Bit	**Reset**	**Read**	**Write**
7 to 6	x	x	x
5	x	x	transfer rate bit 5
4	x	x	transfer rate bit 4
3	x	x	transfer rate bit 3
2	x	x	transfer rate bit 2
1	x	x	transfer rate bit 1
0	x	x	transfer rate bit 0

0213H	IM Bus 1 Master Address Register		
Bit	**Reset**	**Read**	**Write**
7	x	x	IM bus address bit 7
6	x	x	IM bus address bit 6
5	x	x	IM bus address bit 5
4	x	x	IM bus address bit 4
3	x	x	IM bus address bit 3
2	x	x	IM bus address bit 2
1	x	x	IM bus address bit 1
0	x	x	IM bus address bit 0

0214H	IM Bus 1 Master Data Register, Low Byte		
Bit	**Reset**	**Read**	**Write**
7	x	IM bus data bit 7	IM bus data bit 7
6	x	IM bus data bit 6	IM bus data bit 6
5	x	IM bus data bit 5	IM bus data bit 5
4	x	IM bus data bit 4	IM bus data bit 4
3	x	IM bus data bit 3	IM bus data bit 3
2	x	IM bus data bit 2	IM bus data bit 2
1	x	IM bus data bit 1	IM bus data bit 1
0	x	IM bus data bit 0 (LSB)	IM bus data bit 0 (LSB)

0215H	IM Bus 1 Master Data Register, High Byte		
Bit	**Reset**	**Read**	**Write**
7	x	IM bus data bit 15 (MSB)	IM bus data bit 15 (MSB)
6	x	IM bus data bit 14	IM bus data bit 14
5	x	IM bus data bit 13	IM bus data bit 13
4	x	IM bus data bit 12	IM bus data bit 12
3	x	IM bus data bit 11	IM bus data bit 11
2	x	IM bus data bit 10	IM bus data bit 10
1	x	IM bus data bit 9	IM bus data bit 9
0	x	IM bus data bit 8	IM bus data bit 8

0216H	IM Bus 1 Slave 1 Register (IM Bus Address 2)		
Bit	**Reset**	**Read**	**Write**
7	x	IM bus data, bit 7	x
6	x	IM bus data, bit 6	x
5	x	IM bus data, bit 5	x
4	x	IM bus data, bit 4	x
3	x	IM bus data, bit 3	x
2	x	IM bus data, bit 2	x
1	x	IM bus data, bit 1	x
0	x	IM bus data, bit 0	x

0218H	IM Bus 1 Slave 2 Register (IM Bus Address 3)		
Bit	**Reset**	**Read**	**Write**
7	x	IM bus data, bit 7	x
6	x	IM bus data, bit 6	x
5	x	IM bus data, bit 5	x
4	x	IM bus data, bit 4	x
3	x	IM bus data, bit 3	x
2	x	IM bus data, bit 2	x
1	x	IM bus data, bit 1	x
0	x	IM bus data, bit 0	x

021AH	IM Bus 1 Slave 3 Register (IM Bus Address 4)		
Bit	**Reset**	**Read**	**Write**
7	x	IM bus data, bit 7	x
6	x	IM bus data, bit 6	x
5	x	IM bus data, bit 5	x
4	x	IM bus data, bit 4	x
3	x	IM bus data, bit 3	x
2	x	IM bus data, bit 2	x
1	x	IM bus data, bit 1	x
0	x	IM bus data, bit 0	x

021CH	Interrupt Controller Control Register		
Bit	**Reset**	**Read**	**Write**
7 to 5	x	x	x
4	0	x	‘0’ = reset interrupt controller
3	1	x	‘0’ = disable after interrupt
2	1	x	‘0’ = disable interrupts
1	1	x	‘0’ = allow next interrupt
0	0	x	‘0’ = clear all requests

021DH	Interrupt Controller Return Register		
Bit	**Reset**	**Read**	**Write**
7	x	x	x
6	x	x	x
5	x	x	x
4	x	x	x
3	x	x	x
2	x	x	x
1	x	x	x
0	x	x	x

The ‘write’ to this register is the handshake for the interrupt controller that the current interrupt request is served.

021EH	Interrupt Priorities: Source 1 = Timer 2 and Source 0 = Timer 1		
Bit	**Reset**	**Read**	**Write**
7	0	x	x x x x x x x x
6	0	x	0 0 0 0 1 1 1 1
5	0	x	0 0 1 1 0 0 1 1
4	0	x	0 1 0 1 0 1 0 1
3	0	x	x x x x x x x x
2	0	x	0 0 0 0 1 1 1 1
1	0	x	0 0 1 1 0 0 1 1
0	0	x	0 1 0 1 0 1 0 1

Interrupt priority value for source 1 = timer 2
values 0 to 7:
0 = interrupt disabled
1 = lowest priority
7 = highest priority

off low — high

Interrupt priority value for source 0 = timer 1
values 0 to 7:
0 = interrupt disabled
1 = lowest priority
7 = highest priority

021FH	Interrupt Priorities: Source 3 = IM Bus 1 Master and Source 2 = Timer 3		
Bit	**Reset**	**Read**	**Write**
7	0	x	x x x x x x x x
6	0	x	0 0 0 0 1 1 1 1
5	0	x	0 0 1 1 0 0 1 1
4	0	x	0 1 0 1 0 1 0 1
3	0	x	x x x x x x x x
2	0	x	0 0 0 0 1 1 1 1
1	0	x	0 0 1 1 0 0 1 1
0	0	x	0 1 0 1 0 1 0 1

Interrupt priority value for source 3 = IM bus 1 Master
values 0 to 7:
0 = interrupt disabled
1 = lowest priority
7 = highest priority

off low — high

Interrupt priority value for source 2 = timer 3 (1 ms)
values 0 to 7:
0 = interrupt disabled
1 = lowest priority
7 = highest priority

0220H	Interrupt Priorities: Source 5 = IM Bus 2 Master and Source 4 = IM bus 1 Slave			
Bit	**Reset**	**Read**	**Write**	
7	0	x	x x x x x x x x	Interrupt priority value for source 5 = IM bus 2 Master values 0 to 7: 0 = interrupt disabled 1 = lowest priority 7 = highest priority
6	0	x	0 0 0 0 1 1 1 1	
5	0	x	0 0 1 1 0 0 1 1	
4	0	x	0 1 0 1 0 1 0 1	
			off low — high	
3	0	x	x x x x x x x x	Interrupt priority value for source 4 =IM bus 1 slave values 0 to 7: 0 = interrupt disabled 1 = lowest priority 7 = highest priority
2	0	x	0 0 0 0 1 1 1 1	
1	0	x	0 0 1 1 0 0 1 1	
0	0	x	0 1 0 1 0 1 0 1	

0221H	Interrupt Priorities: Source 7 = P87 and Source 6 = IM bus 2 Slave			
Bit	**Reset**	**Read**	**Write**	
7	0	x	x x x x x x x x	Interrupt priority value for source 7 = P87 values 0 to 7: 0 = interrupt disabled 1 = lowest priority 7 = highest priority
6	0	x	0 0 0 0 1 1 1 1	
5	0	x	0 0 1 1 0 0 1 1	
4	0	x	0 1 0 1 0 1 0 1	
			off low — high	
3	0	x	x x x x x x x x	Interrupt priority value for source 4 =IM bus 2 slave values 0 to 7: 0 = interrupt disabled 1 = lowest priority 7 = highest priority
2	0	x	0 0 0 0 1 1 1 1	
1	0	x	0 0 1 1 0 0 1 1	
0	0	x	0 1 0 1 0 1 0 1	

0222H	Timer 1 Control Register 1		
Bit	**Reset**	**Read**	**Write**
7	x	x	‘1’ = second serial input level enabled
6	x	x	clock select, bit 1: ‘00’ = pin, ‘01’ = fosc
5	x	x	clock select, bit 0: ‘10’ = PHI2, ‘11’ = no clock
4	x	x	counter stop: ‘0’ = disabled, ‘1’ = carry out accu
3	x	x	start condition, bit 1: ‘00’ = none (always active)
2	x	x	start condition, bit 0: ‘01’ = edge, ‘10’ = pin, ‘11’ = \ pin
1	x	x	‘1’ = half load enabled
0	x	x	active edge selection: ‘0’ = rising, ‘1’ = falling

0223H	Timer 1 Control Register 2		
Bit	**Reset**	**Read**	**Write**
7	x	x	pin output mode, bit 1: 00 = disabled, 01 = serial out
6	x	x	pin output mode, bit 0: 10 = PWM, 11 = carry accu D
5	x	x	read latch, bit 2: 000 = disabled, 001 = carry accu C
4	x	x	read latch, bit 1: ‘010’ = carry accu D, ‘011’ pin
3	x	x	read latch, bit 0: ‘100’ = \pin, ‘101’ = prescaler output, ‘110’ = prescaler input, ‘111’ = undefined
2	x	x	accu clock, bit 1: ‘00’ = presc. input, ‘01’ = pre.output
1	x	x	accu clock, bit 0: ‘10’ = pin, ‘11’ = \ pin
0	x	x	‘1’ = use accu c with accu d as long one (16bit accu)

0224H	Timer 1 Control Register 3		
Bit	**Reset**	**Read**	**Write**
7	x	x	interrupt event, bit 2:‘000’ = none, ‘001’ = pin
6	x	x	interrupt event, bit 1:‘010’ = \pin, ‘011’ = carry accu C,
5	x	x	interrupt event, bit 0:‘100’ = carry accu D, ‘101’, ‘110’, ‘111’ = undefined
4	x	x	load event, bit 1: ‘00’: ‘00’ = none, ‘01’ = carry accu C,
3	x	x	load event, bit 0: ‘10 = carry accu D, ‘11’ = reg. load
2	x	x	accu C input: ‘0’ = bus register, ‘1’ = –1
1	x	x	accu D input: ‘0’ = bus register, ‘1’ = –1
0	x	x	serial mode enable: ‘0’ = disable, ‘1’ = enable

0225H	Timer 1 Prescaler Low Byte		
Bit	**Reset**	**Read**	**Write**
7	x	x	scaler value bit 7
6	x	x	scaler value bit 6
5	x	x	scaler value bit 5
4	x	x	scaler value bit 4
3	x	x	scaler value bit 3
2	x	x	scaler value bit 2
1	x	x	scaler value bit 1
0	x	x	scaler value bit 0 (LSB)

0226H	Timer 1 Prescaler High Byte		
Bit	**Reset**	**Read**	**Write**
7	x	x	scaler value bit 15 (MSB)
6	x	x	scaler value bit 14
5	x	x	scaler value bit 13
4	x	x	scaler value bit 12
3	x	x	scaler value bit 11
2	x	x	scaler value bit 10
1	x	x	scaler value bit 9
0	x	x	scaler value bit 8

0228H	Timer 1 Accu Low Byte (Accu C)		
Bit	**Reset**	**Read**	**Write**
7	x	x	accu bit 7
6	x	x	accu bit 6
5	x	x	accu bit 5
4	x	x	accu bit 4
3	x	x	accu bit 3
2	x	x	accu bit 2
1	x	x	accu bit 1
0	x	x	accu bit 0 (LSB)

0229H	Timer 1 Accu High Byte (Accu D)		
Bit	**Reset**	**Read**	**Write**
7	x	x	accu bit 15 (MSB)
6	x	x	accu bit 14
5	x	x	accu bit 13
4	x	x	accu bit 12
3	x	x	accu bit 11
2	x	x	accu bit 10
1	x	x	accu bit 9
0	x	x	accu bit 8

022AH	Timer 1 Adder Low Byte		
Bit	**Reset**	**Read**	**Write**
7	x	adder bit 7	adder bit 7
6	x	adder bit 6	adder bit 6
5	x	adder bit 5	adder bit 5
4	x	adder bit 4	adder bit 4
3	x	adder bit 3	adder bit 3
2	x	adder bit 2	adder bit 2
1	x	adder bit 1	adder bit 1
0	x	adder bit 0 (LSB)	adder bit 0 (LSB)

022BH	Timer 1 Adder High Byte		
Bit	**Reset**	**Read**	**Write**
7	x	adder bit 15 (MSB)	adder bit 15 (MSB)
6	x	adder bit 14	adder bit 14
5	x	adder bit 13	adder bit 13
4	x	adder bit 12	adder bit 12
3	x	adder bit 11	adder bit 11
2	x	adder bit 10	adder bit 10
1	x	adder bit 9	adder bit 9
0	x	adder bit 8	adder bit 8

022CH	Timer 2 Control Register 1		
Bit	**Reset**	**Read**	**Write**
7	x	x	‘1’ = second serial input level enabled
6	x	x	clock select, bit 1: ‘00’ = pin, ‘01’ = fosc.,
5	x	x	clock select, bit 0: ‘10’ = PHI2, ‘11’ = no clock
4	x	x	counter stop: ‘0’ = disabled , ‘1’ = carry out accu
3	x	x	start condition, bit 1: ‘00’= none (always active),
2	x	x	start condition, bit 0: ‘01’ = edge, ‘10’ = pin, ‘11’ =\pin
1	x	x	‘1’ = half load enabled
0	x	x	active edge selection: ‘0’ = rising ‘1’ = falling

022DH	Timer 2 Control Register 2		
Bit	**Reset**	**Read**	**Write**
7	x	x	pin output mode, bit 1: ‘00’ = disabled, ‘01’ = serial out,
6	x	x	pin output mode, bit 0: ‘10’ = PWM, ‘11’ = carry accu D
5	x	x	read latch, bit 2: ‘000’ = disabled, ‘001’ = carry accu C
4	x	x	read latch, bit 1: ‘010’ = carry accu D, ‘011’ pin
3	x	x	read latch, bit 0: ‘100’ = \pin, ‘101’ = prescaler output, ‘110’ = presc. input, ‘111’ = undefined
2	x	x	accu clock, bit 1: ‘00’ = presc. input, ‘01’ = presc. output
1	x	x	accu clock, bit 0: ‘10’ = pin, ‘11’ = \pin
0	x	x	‘1’ = use accu C with accu D as long one (16-bit accu)

022EH	Timer 2 Control Register 3		
Bit	**Reset**	**Read**	**Write**
7	x	x	interrupt event, bit 2: ‘000’ = none, ‘001’ = pin
6	x	x	interrupt event, bit 1: ‘010’ = \pin, ‘011’ = carry accu C
5	x	x	interrupt event, bit 0: ‘100’ = carry accu D, ‘101’, ‘110’, ‘111’ = undefined
4	x	x	load event, bit 1: ‘00’ = none, ‘01’ = carry accu C,
3	x	x	load event, bit 0: ‘10’ = carry accu D, ‘11’ = register load
2	x	x	accu C input: ‘0’ = bus register, ‘1’ = –1
1	x	x	accu D input: ‘0’ = bus register, ‘1’ = –1
0	x	x	serial mode enable: ‘0’ = disable, ‘1’ = enable

022FH	Timer 2 Prescaler Low Byte		
Bit	**Reset**	**Read**	**Write**
7	x	x	scaler value bit 7
6	x	x	scaler value bit 6
5	x	x	scaler value bit 5
4	x	x	scaler value bit 4
3	x	x	scaler value bit 3
2	x	x	scaler value bit 2
1	x	x	scaler value bit 1
0	x	x	scaler value bit 0 (LSB)

0230H	Timer 2 Prescaler High Byte		
Bit	**Reset**	**Read**	**Write**
7	x	x	scaler value bit 15 (MSB)
6	x	x	scaler value bit 14
5	x	x	scaler value bit 13
4	x	x	scaler value bit 12
3	x	x	scaler value bit 11
2	x	x	scaler value bit 10
1	x	x	scaler value bit 9
0	x	x	scaler value bit 8

0232H	Timer 2 Accu Low Byte (Accu C)		
Bit	**Reset**	**Read**	**Write**
7	x	x	accu bit 7
6	x	x	accu bit 6
5	x	x	accu bit 5
4	x	x	accu bit 4
3	x	x	accu bit 3
2	x	x	accu bit 2
1	x	x	accu bit 1
0	x	x	accu bit 0 (LSB)

0233H	Timer 2 Accu High Byte (Accu D)		
Bit	**Reset**	**Read**	**Write**
7	x	x	accu bit 15 (MSB)
6	x	x	accu bit 14
5	x	x	accu bit 13
4	x	x	accu bit 12
3	x	x	accu bit 11
2	x	x	accu bit 10
1	x	x	accu bit 9
0	x	x	accu bit 8

0234H	Timer 2 Adder Low Byte		
Bit	**Reset**	**Read**	**Write**
7	x	adder bit 7	adder bit 7
6	x	adder bit 6	adder bit 6
5	x	adder bit 5	adder bit 5
4	x	adder bit 4	adder bit 4
3	x	adder bit 3	adder bit 3
2	x	adder bit 2	adder bit 2
1	x	adder bit 1	adder bit 1
0	x	adder bit 0 (LSB)	adder bit 0 (LSB)

0235H	Timer 2 Adder High Byte		
Bit	**Reset**	**Read**	**Write**
7	x	adder bit 15 (MSB)	adder bit 15 (MSB)
6	x	adder bit 14	adder bit 14
5	x	adder bit 13	adder bit 13
4	x	adder bit 12	adder bit 12
3	x	adder bit 11	adder bit 11
2	x	adder bit 10	adder bit 10
1	x	adder bit 9	adder bit 9
0	x	adder bit 8	adder bit 8

0236H	Timer 3 Control Register 1		
Bit	**Reset**	**Read**	**Write**
7	x	x	‘1’ = second serial input level enabled
6	x	x	clock select, bit 1: ‘00’ = pin, ‘01’ = fosc.,
5	x	x	clock select, bit 0: ‘10’ = PHI2, ‘11’ = no clock
4	x	x	counter stop: ‘0’ = disabled , ‘1’ = carry out accu
3	x	x	start condition, bit 1: ‘00’= none (always active),
2	x	x	start condition, bit 0: ‘01’ = edge, ‘10’ = pin, ‘11’ =\pin
1	x	x	‘1’ = half load enabled
0	x	x	active edge selection: ‘0’ = rising, ‘1’ = falling

0237H	Timer 3 Control Register 2		
Bit	**Reset**	**Read**	**Write**
7	x	x	pin output mode, bit 1: 00 = disabled, 01 = serial out
6	x	x	pin output mode, bit 0: 10 = PWM, 11 = carry accu D
5	x	x	read latch, bit 2: 000 = disabled, 001 = carry accu C
4	x	x	read latch, bit 1: ‘010’ = carry accu D, ‘011’ pin
3	x	x	read latch, bit 0: ‘100’ = \pin, ‘101’ = prescaler out-put, ‘110’ = presc. input, ‘111’ = undefined
2	x	x	accu clock, bit 1: ‘00’ = presc. input, ‘01’ = presc. output
1	x	x	accu clock, bit 0: ‘10’ = pin, ‘11’ = \pin
0	x	x	1 = use acc. C with acc. D as long one (16-bit acc.)

0238H	Timer 3 Control Register 3		
Bit	**Reset**	**Read**	**Write**
7	x	x	interrupt event, bit 2: ‘000’ = none, ‘001’ = pin
6	x	x	interrupt event, bit 1: ‘010’ = \pin, 011 = carry accu C
5	x	x	interrupt event, bit 0: ‘100’ = carry accu D, ‘101’, ‘110’, ‘111’ = undefined
4	x	x	load event, bit 1: ‘00’ = none, ‘01’ = carry accu C,
3	x	x	load event, bit 0: 10 = carry accu D, 11 = regist. load
2	x	x	accu C input: ‘0’ = bus register, ‘1’ = –1
1	x	x	accu D input: ‘0’ = bus register, ‘1’ = –1
0	x	x	serial mode enable: ‘0’ = disable, ‘1’ = enable

0239H	Timer 3 Prescaler Low Byte		
Bit	**Reset**	**Read**	**Write**
7	x	x	scaler value bit 7
6	x	x	scaler value bit 6
5	x	x	scaler value bit 5
4	x	x	scaler value bit 4
3	x	x	scaler value bit 3
2	x	x	scaler value bit 2
1	x	x	scaler value bit 1
0	x	x	scaler value bit 0 (LSB)

023AH	Timer 3 Prescaler High Byte		
Bit	**Reset**	**Read**	**Write**
7	x	x	scaler value bit 15 (MSB)
6	x	x	scaler value bit 14
5	x	x	scaler value bit 13
4	x	x	scaler value bit 12
3	x	x	scaler value bit 11
2	x	x	scaler value bit 10
1	x	x	scaler value bit 9
0	x	x	scaler value bit 8

023CH	Timer 3 Accu Low Byte (Accu C)		
Bit	**Reset**	**Read**	**Write**
7	x	x	accu bit 7
6	x	x	accu bit 6
5	x	x	accu bit 5
4	x	x	accu bit 4
3	x	x	accu bit 3
2	x	x	accu bit 2
1	x	x	accu bit 1
0	x	x	accu bit 0 (LSB)

023DH	Timer 3 Accu High Byte (Accu D)		
Bit	**Reset**	**Read**	**Write**
7	x	x	accu bit 15 (MSB)
6	x	x	accu bit 14
5	x	x	accu bit 13
4	x	x	accu bit 12
3	x	x	accu bit 11
2	x	x	accu bit 10
1	x	x	accu bit 9
0	x	x	accu bit 8

023EH	Timer 3 Adder Low Byte		
Bit	**Reset**	**Read**	**Write**
7	x	adder bit 7	adder bit 7
6	x	adder bit 6	adder bit 6
5	x	adder bit 5	adder bit 5
4	x	adder bit 4	adder bit 4
3	x	adder bit 3	adder bit 3
2	x	adder bit 2	adder bit 2
1	x	adder bit 1	adder bit 1
0	x	adder bit 0 (LSB)	adder bit 0 (LSB)

023FH	Timer 3 Adder High Byte		
Bit	**Reset**	**Read**	**Write**
7	x	adder bit 15 (MSB)	adder bit 15 (MSB)
6	x	adder bit 14	adder bit 14
5	x	adder bit 13	adder bit 13
4	x	adder bit 12	adder bit 12
3	x	adder bit 11	adder bit 11
2	x	adder bit 10	adder bit 10
1	x	adder bit 9	adder bit 9
0	x	adder bit 8	adder bit 8

0240H	Port 6 Data Register		
Bit	**Reset**	**Read**	**Write**
7 to 0	0	Port 6 Data	Port 6 Data

0241H	Port 6 Direction Register		
Bit	**Reset**	**Read**	**Write**
7 to 0	1	x	‘1’ = input, ‘0’ = output mode

0242H	Port 7 Data Register		
Bit	**Reset**	**Read**	**Write**
7 to 0	0	Port 7 Data	Port 7 Data

0243H	Port 7 Direction Register		
Bit	**Reset**	**Read**	**Write**
7 to 0	1	x	‘1’ = input, ‘0’ = output mode

0244H	Port 8 Data Register		
Bit	**Reset**	**Read**	**Write**
7	0	Data bit 7	Data bit 7
6 to 4	x	x	x
3	0	Data bit 3	Data bit 3
2	0	Data bit 2	Data bit 2
1	0	Data bit 1	Data bit 1
0	0	Data bit 0	Data bit 0

0245H	Port 8 Direction Register		
Bit	**Reset**	**Read**	**Write**
7	1	x	‘1’ = input, ‘0’ = output mode
6	x	x	x
5	x	x	x
4	x	x	x
3	1	x	‘1’ = input, ‘0’ = output mode
2	1	x	‘1’ = input, ‘0’ = output mode
1	1	x	‘1’ = input, ‘0’ = output mode
0	1	x	‘1’ = input, ‘0’ = output mode

0246H	IM Bus 2 Control and Status Register			
Bit	**Reset**	**Read**	**Write**	
7 to 4	x	x	x	
3	0	‘1’ = 1 byte received in slave register 3 (IM bus address 4)	‘1’ = read word via IM bus (master)	(bits 0 ... 3: ‘0000’= reset IM bus interface)
2	0	‘1’ = 1 byte received in slave register 2 (IM bus address 3)	‘1’ = read byte via IM bus (master)	
1	0	‘1’ = 1 byte received in slave register 1 (IM bus address 2)	‘1’ = write word via IM bus (master)	
0	0	‘1’ = IM bus (master) busy	‘1’ = write byte via IM bus (master)	

0247H	IM Bus 2 Data Transfer Rate Register		
Bit	**Reset**	**Read**	**Write**
7 to 6	x	x	x
5	x	x	transfer rate bit 5
4	x	x	transfer rate bit 4
3	x	x	transfer rate bit 3
2	x	x	transfer rate bit 2
1	x	x	transfer rate bit 1
0	x	x	transfer rate bit 0

0249H	IM Bus 2 Master Address Register		
Bit	**Reset**	**Read**	**Write**
7	x	x	IM bus address bit 7
6	x	x	IM bus address bit 6
5	x	x	IM bus address bit 5
4	x	x	IM bus address bit 4
3	x	x	IM bus address bit 3
2	x	x	IM bus address bit 2
1	x	x	IM bus address bit 1
0	x	x	IM bus address bit 0

024AH	IM Bus 2 Master Data Register, Low Byte		
Bit	**Reset**	**Read**	**Write**
7	x	IM bus data, low byte, bit 7	IM bus data, low byte, bit 7
6	x	IM bus data, low byte, bit 6	IM bus data, low byte, bit 6
5	x	IM bus data, low byte, bit 5	IM bus data, low byte, bit 5
4	x	IM bus data, low byte, bit 4	IM bus data, low byte, bit 4
3	x	IM bus data, low byte, bit 3	IM bus data, low byte, bit 3
2	x	IM bus data, low byte, bit 2	IM bus data, low byte, bit 2
1	x	IM bus data, low byte, bit 1	IM bus data, low byte, bit 1
0	x	IM bus data, low byte, bit 0	IM bus data, low byte, bit 0

024BH	IM Bus 2 Master Data Register, High Byte		
Bit	**Reset**	**Read**	**Write**
7	x	IM bus data, high byte, bit 7	IM bus data, high byte, bit 7
6	x	IM bus data, high byte, bit 6	IM bus data, high byte, bit 6
5	x	IM bus data, high byte, bit 5	IM bus data, high byte, bit 5
4	x	IM bus data, high byte, bit 4	IM bus data, high byte, bit 4
3	x	IM bus data, high byte, bit 3	IM bus data, high byte, bit 3
2	x	IM bus data, high byte, bit 2	IM bus data, high byte, bit 2
1	x	IM bus data, high byte, bit 1	IM bus data, high byte, bit 1
0	x	IM bus data, high byte, bit 0	IM bus data, high byte, bit 0

024CH	IM Bus 2 Slave 1 Register (IM Bus Address 2)		
Bit	**Reset**	**Read**	**Write**
7	x	IM bus data, bit 7	x
6	x	IM bus data, bit 6	x
5	x	IM bus data, bit 5	x
4	x	IM bus data, bit 4	x
3	x	IM bus data, bit 3	x
2	x	IM bus data, bit 2	x
1	x	IM bus data, bit 1	x
0	x	IM bus data, bit 0	x

024EH	IM Bus 2 Slave 2 Register (IM Bus Address 3)		
Bit	**Reset**	**Read**	**Write**
7	x	IM bus data, bit 7	x
6	x	IM bus data, bit 6	x
5	x	IM bus data, bit 5	x
4	x	IM bus data, bit 4	x
3	x	IM bus data, bit 3	x
2	x	IM bus data, bit 2	x
1	x	IM bus data, bit 1	x
0	x	IM bus data, bit 0	x

0250H	IM Bus 2 Slave 3 Register (IM Bus Address 4)		
Bit	**Reset**	**Read**	**Write**
7	x	IM bus data, bit 7	x
6	x	IM bus data, bit 6	x
5	x	IM bus data, bit 5	x
4	x	IM bus data, bit 4	x
3	x	IM bus data, bit 3	x
2	x	IM bus data, bit 2	x
1	x	IM bus data, bit 1	x
0	x	IM bus data, bit 0	x

5. Index

6. Addendum: CCU 3000, CCU 3000-I EMU Versions

The CCU 3000 TCs 10, 12, 16, 1, and CCU 3000-I TCs 1 and 3 are emulator versions (EMUs). They differ from production versions in the programmability of control register bit 5: If this bit is set to 0, the CCU assumes to have a clock signal at its X1-pin instead of a crystal connected at pins X1 and X2. X2, in that case, works as a clock output delivering the inverted processor Φ2 output signal.

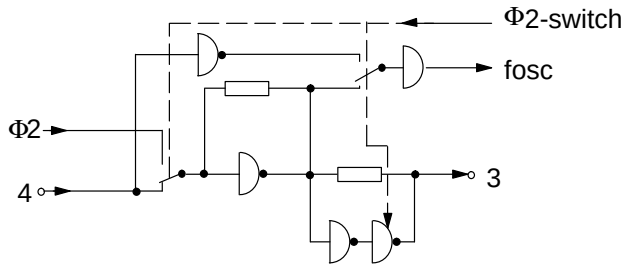


Fig. 6–1: X1, X2
Position shown:
Φ2-switch = 1
XTAL-mode

0201H	**Control Register**		
Bit	**Reset**	**Read**	**Write**
7	copy from addr. FFF9H	x	no function – set to ‘1’ (to keep compatibility)
6		x	no function – set to ‘1’ (to keep compatibility)
5		PHI2-out: ‘0’ = active, ‘1’ = inactive	PHI2-out: ‘0’ = active, ‘1’ = inactive
4		external bus: ‘1’ = bus on ports 0, 1, 2 disabled	Bus disable: ‘1’ = disable bus on ports 0, 1, 2
3		R/W signal / Port4: ‘0’ = R/W, ‘1’ = P40	R/W signal / Port4: ‘0’ = R/W, ‘1’ = P40
2		internal ROM: ‘1’ = internal ROM enabled	ROM enable: ‘1’ = enable internal ROM
1		internal RAM: ‘1’ = internal RAM enabled	RAM enable: ‘1’ = enable internal RAM
0		internal CPU: ‘1’ = internal CPU enabled	CPU enable: ‘1’ = enable internal CPU

7. Addendum: CCU 3000 1 µm Version

As the 1 µm version of the CCU has faster pin signal drivers than the 1.2 µm version, it may be programmed to work in a kind of “slow mode”, i.e.: the current of the pin driver transistors in that mode is limited to 35 to 40%. In this default mode it is compatible with the 1.2 µm version.

020EH	**Fast/Slow Register**		
Bit	**Reset**	**Read**	**Write**
7 to 3	x	x	x
2	0	x	Port 6: 0 = slow, 1 = fast
1	0	x	Ports 5, 7, 8, Timers 1, 2, 3, IR, IM bus: 0 = slow, 1 = fast
0	0	x	Ports 1, 2, 3, 4,: 0 = slow, 1 = fast

7.1. Electrical Characteristics

7.1.1. Absolute Maximum Ratings

Symbol	Parameter	Pin No.	Min.	Max.	Unit	Test Conditions
Voltages						
V_{DD}	Supply Voltage	1	–0.5	6.5	V	
V_I	Input Voltage on any pin	1 to 68	–0.3	V_{DD}+0.3	V	
Currents						
I_{DD}	Supply Current	1	–50	80	mA	
I_{STBY}	Standby Current	5	–280	50	mA	
I_I	Input Current	7 to 68	–2	2	mA	
I_O	Output Current	7 to 60	–5	5	mA	(except Port 6)
I_O	Output Current	61 to 68		30	mA	(Port 6 only)
Temperatures						
T_A	Ambient Temperature under Bias	–	–10	80	°C	
T_S	Storage Temperature	–	–40	125	°C	
Power						
P_{max}	Power Dissipation	–		1210 800	mW mW	$T_A \leq 70$ °C, 68-pin PLCC $T_A \leq 70$ °C, 64-pin SDIP

Stresses beyond those listed in the “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only. Functional operation of the device at these or any other conditions beyond those indicated in the “Recommended Operating Conditions/Characteristics” of this specification is not implied. Exposure to absolute maximum ratings conditions for extended periods may affect device reliability.

7.1.2. Recommended Operating Conditions at V_{DD} = 4.75 V, T_{AMB} = 0 °C to 70°C

Symbol	Parameter	Pin No.	Min.	Typ.	Max.	Unit	Test Conditions
Input Voltages							
V_{INL}	Input Low Voltage	3, 4, 6–68	V_{ss} – 0.3	–	0.8	V	
V_{IH}	Input High Voltage		2.8	–	V_{DD} +0.3	V	
Open Drain Output Voltages							
V_{OH}	Output High Voltage	61–68		–	V_{DD} +0.3	V	(Port 6 only)
Standby Voltage							
V_{STBY}		5	3	–	5.25	V	
Clock Input							
F_{XTAL}		4	0.5	–	8	MHz	external clock
Capacitive Load on Address, RWQ and Data Pads (Port1, Port2, Port3, Port4)							
C_{ADF}		17–41			30	pF	Bus fast mode <020E>, Bit0=0
C_{ADS}					100	pF	Bus slow mode <020E>, Bit0=1

7.1.3. Recommended Crystal Characteristics at $C_{XTAL1} = C_{XTAL2}$ = 22 pF±1p; $C_{stray} \leq$ 2 pF

Symbol	Parameter	Pin No.	Min.	Typ.	Max.	Unit	Test Conditions
Quartz (XTAL1/XTAL2)							
f_p	Parallel Resonance Frequency	–	–	4–8	–	MHz	$C_{L=}$ 13 pF
R_1	Series Resistance	–	–	–	40 150	Ω Ω	8 MHz 4 MHz
C_0	Shunt Capacitance	–	–	–	7.0	pF	
C_1	Motional Capacitance	–	–	–	20	fF	
P	Rated Drive Level	–	–	0.02	–	mW	
f_p / f_H	Spurious Frequency Attenuation	–	20	–	–	dB	

7.1.4. DC Characteristics at V_{DD} = 4.75V to 5.25V, T_{AMB} = 0 °C to 70°C,f_{XTAL} = 8 MHz, for 68-Pin PLCC Package

Symbol	Parameter	Pin No.	Min.	Typ.	Max.	Unit	Comment
I_{DD}	Supply Current (no external load, CMOS-levels on inputs)	1	–	8/16	15/30	mA	@ 4/8 MHz
I_{STBY}	Standby Current	5			1	µA	V_{DD}=0V, V_{STBY} =5V
V_{POR}	Power-On Reset Voltage	6	4	4.2	4.75	V	
Inputs (all inputs except XTAL1)							
V_{ILH}	Schmitt Input L → H Transition Voltage	3, 6–68	0.38 V_{DD}	2.15	0.56 V_{DD}	V	
V_{IHL}	Schmitt Input H → L Transition Voltage	3, 6–68	0.20 V_{DD}	1.15	0.29 ·V_{DD}	V	
Δ (V_{ILH}–V_{IHL})	Schmitt Input Hysteresis		0.10 V_{DD}		0.27 ·V_{DD}	V	
I_{LI}	Leakage Current		–1		1	µA	$V_{ss} \le V_{IN} \le V_{DD}$
Outputs (push-pull or push-pull/open-drain switch: P1..5, P7..8, T1..3, IR, XTAL2)							
V_{OL}	Output Low Voltage	17–60			0.4	V	I_{OUT}=4mA
V_{OH}	Output High Voltage		V_{DD} – 0.4			V	I_{OUT}=–4mA
Outputs (open-drain with weak pull-up: IMB_ID1, IMB_DAT1, IMB_CL1, IMB_ID2, IMB_DAT2, IMB_CL2, RESQ)							
V_{OL}	Output Low Voltage	17–60			0.4	V	I_{OUT} = 4 mA
V_{OH}	Output High Voltage		V_{DD} – 0.4			V	I_{OUT} = –2 µA
I_{OHS}	Output High Short Circuit Current		–50			µA	V_{OUT} = V_{SS} Output data = 1
Outputs (open drain, with clamping diode: P6)							
V_{OL}	Output Low Voltage	61–68			0.5	V	I_{OUT}= 25 mA

7.1.5. AC Characteristics at T_{AMB} = 0 °C to 70°C, V_{DD} = 4.75 V to 5.25 V, Cl = 0 pF
External Loads: add 0.75 ns/pF for controller output lines

Symbol	Parameter	Pin	Min.	Max.	Unit
t_{cyc}	Cycle Time	3	125	2000	ns
t_{PWL}	Pulse Width Low	3	60	1000	ns
t_{AH}	Address Hold Time	26–41	10	22	ns
t_{ADS}	Address Setup Time		15	34	ns
t_{DSR}	Data Setup Time Read	18–25	20		ns
t_{MDS}	Write Data Delay		10	29	ns
t_{DHW}	Write Data Hold Time		9	16	ns
t_{DHR}	Read Data Hold Time		10		ns
t_{RWH}	Read/$\overline{\text{Write}}$ Hold Time	17	10	24	ns
t_{WRH}	$\overline{\text{Read}}$/Write Hold Time	17	13	34	ns
t_{P5S}	Delay Port1 to Port3 Select Lines	42–47	12	26	ns
t_{X1PH2}	Delay X1 to internal Φ2	4	7	15	ns
t_{PH2X2}	Internal Φ2 to X2 Output	3	5	10	ns

8. Addendum: CCU 3000-I Specification

8.1. Changes to CCU3000

Instead of the Master/Slave IM bus Interface IM1 of CCU3000, an I^2C/IM bus Master is used.

Source 3 of the interrupt controller is not connected. Its priority has to be set to '0'.

Source 4 of the interrupt controller is connected with Port81 (Special Input). Falling edges of port 8, bit 1 generate interrupts if the priority of this interrupt source is not set to 0. In I^2C mode it is possible to switch I^2C_CLK from IM1_CLK_Pad to IM1_ID_Pad. Therefore two I^2C busses can be driven (see section 8.6. for details).

The CCU 3000-I is available in two different packages, see pages 71 to 73. All other features are the same as in CCU3000.

8.2. Definitions

8.3. Interrupt Definitions

Interrupt	Source	Vector (low, high byte)
0	TIMER1	FFF6, FFF7
1	TIMER2	FFF4, FFF5
2	TIMER3	FFF2, FFF3
3	NC	FFF0, FFF1
4	P81	FFEE, FFEF
5	IM-BUS2, Master	FFEC, FFED
6	IM-BUS2, Slave	FFEA, FFEB
7	P87	FFE8, FFE9
RESET		FFFC, FFFD

8.4. Memory Mappings

RAM	0000H to 01FFH	Page 0, 1
	0300H to 063FH	Page 3, 4, 5, 6
ROM	8000H to FFFFH	(CCU3001-I only)
Control byte	FFF9	
I/O	0200 to 02FF	

8.5. I/O Definitions

Address	Function
200H	Clock frequency
201H	Control register
202H	Watchdog
203H	Port 1 Data
204H	Direction Register Port 1
205H	Port 2 Data
206H	Direction Register Port 2
207H	Port 3 Data
208H	Direction Register Port 3
209H	Port 4 Data
20AH	Port 5 Mode Register
20BH	Port 5 Direction Register
20CH	Port 5 Data
20DH	IR-Input
20FH	Port 7 Mode Register
21CH	Interrupt controller control byte
21DH	Interrupt controller return byte
21EH	Interrupt controller priorities source 0 & 1
21FH	Interrupt controller priorities source 2 & 3
220H	Interrupt controller priorities source 4 & 5
221H	Interrupt controller priorities source 6 & 7
222H	Timer 1 control byte 1
223H	Timer 1 control byte 2
224H	Timer 1 control byte 3
225H	Timer 1 prescaler low byte
226H	Timer 1 prescaler high byte
228H	Timer 1 accu low byte
229H	Timer 1 accu high byte
22AH	Timer 1 adder low byte
22BH	Timer 1 adder high byte
22CH	Timer 2 control byte 1
22DH	Timer 2 control byte 2
22EH	Timer 2 control byte 3
22FH	Timer 2 prescaler low byte
230H	Timer 2 prescaler high byte
232H	Timer 2 accu low byte
233H	Timer 2 accu high byte
234H	Timer 2 adder low byte
235H	Timer 2 adder high byte
236H	Timer 3 control byte 1
237H	Timer 3 control byte 2
238H	Timer 3 control byte 3
239H	Timer 3 prescaler low byte
23AH	Timer 3 prescaler high byte
23CH	Timer 3 accu low byte
23DH	Timer 3 accu high byte
23EH	Timer 3 adder low byte
23FH	Timer 3 adder high byte
240H	Port 6 Data
241H	Direction Register Port 6
242H	Port 7 Data
243H	Direction Register Port 7
244H	Port 8 Data
245H	Direction Register Port 8
246H	IM-Bus 2 control & status
247H	IM-Bus 2 transfer rate
249H	IM-Bus 2 master address
24AH	IM-Bus 2 master data low
24BH	IM-bus 2 master data high
24CH	IM-bus 2 slave 1, IM address 02
24EH	IM-bus 2 slave 2, IM address 03

250H	IM-bus 2 slave 3, IM address 04
2D0H	I^2C Start Cycle without generation of ACK (ACK = '1')
2D1H	I^2C Start Cycle with generation of ACK (ACK = '0')
2D2H	I^2C Resume Cycle without generation of ACK (ACK = '1')
2D3H	I^2C Resume Cycle with generation of ACK (ACK = '0')
2D4H	I^2C Termination Cycle without generation of ACK (ACK = '1')
2D5H	I^2C Termination Cycle with generation of ACK (ACK = '0')
2D6H	I^2C / IM bus Data from Receive-FIFO
2D7H	I^2C / IM bus Status
2D8H	IM bus Start Cycle
2D9H	IM bus Resume Cycle
2DAH	IM bus Termination Cycle
2DBH	I^2C / IM bus Prescaler
2E0H to 2E7H	External addresses, used for EMU boards
2FEH	Reserved, do not use
2FFH	Reserved for testing purposes

8.6. I^2C and IM Bus Interface

The master bus interface can generate two different kinds of format:

– I^2C format

– IM format

The MSBit of the bus prescaler registers (address 2DBH) is used to switch I^2C_CLK between IM1_CLK_Pad and IM1_ID_Pad. The remaining 7 bits can be used to set the bit rate.

bit 7	0 = I^2C_CLK at IM1_ID_Pad, 1 = I^2C_CLK at IM1_CLK_Pad (reset state)
bit 6 to 0	bit rate $f_{IMI2C} = f_{OSC} / (4 * n)$ for n>1

where n is the value of bits 0 to 6 and the setting value (0 = reset state means n = 128). A complete telegram is assembled by the software out of individual sections. Each section contains an 8-bit data. This data is written into one of the nine possible Control-Data registers. Depending on the chosen address, a certain part of an I^2C or IM bus cycle is generated. By means of corresponding calling sequences it is therefore possible to join even very long telegrams (e.g. long data files for auto increment addressing of I^2C slaves).

The software interface contains a 3 byte deep FIFO for the control-data registers as well as for the received data. Thus all IM and most of the I^2C telegrams can be transmitted to the hardware without the software having to wait for empty space in the FIFO.

All address and data fields appearing on the bus are constantly read and written into the Read-FIFO. The software can then check these data in comparison with the scheduled data. If a read instruction is handled, the interface must set the data word FFH so that the responding slave can insert its data. In this case the Read-FIFO contains the read-in data.

If telegrams longer than 3 bytes are received, (1 address, 2 data bytes), the software must check the filling condition of the control data FIFO and, if necessary, fill it up (or read out the Read-FIFO). A variety of status flags is available for this purpose:

– The 'half-full' flag is set if there are more than two bytes available in the Transmit-FIFO.

– Bus Busy is activated by writing any byte to any one of the data transfer registers. It stays active until the I^2C or IM bus activities are stopped after the stop condition generation. So 'Busy' becomes inactive after the data that was written in one of the four registers to terminate the bus action is completely shifted out, and the bus-specific stop condition is generated (see Fig. 2–22, 2–25).

Moreover, in the I^2C mode the ACK-bit is recorded separately on the bus lines for the address and the data fields; however, the interface itself can set the address ACK=0. In any case the two ACK flags show the actual bus condition. These flags remain until the next I^2C start condition is generated.

Table 2–1: I²C and IM bus interface registers

Address	*Function*	
2D0H(w)	generate I²C start condition, transfer Data as I²C address, and set ACK=1	
2D1H(w)	same as above, ACK=0	
2D2H(w)	output 8 I²C Data bits, set ACK=1	
2D3H(w)	same as above, set ACK=0	
2D4H(w)	output 8 I²C Data bits, set ACK=1, generate I²C stop condition	
2D5H(w)	same as above, set ACK=0	
2D6H(r)	receive FIFO	
2D7H(r)	status flags:	
	bit 0	not used
	bit 1	1= receive FIFO empty
	bit 2	1= contr-data-FIFO half full
	bit 3	1= Bus busy
	bit 4	I²C data ACK
	bit 5	I²C adr ACK
	bit 6	“OR”ed ACK
	bit 7	not used
2D8H(w)	generate IM-address field	
2D9H(w)	generate 8 IM-data bits	
2DAH(w)	generate 8 IM-data bits and the IM-stop condition	
2DBH(w)	terminal select & speed	

For example, the software has to work off the following sequence (ACK =1) to read a 16-bit word from an I²C device address 10H (on condition that the bus is not active):

–write 21H to	2D0H	
–write 0FFH to	2D2H	
–write 0FFH to	2D4H	check receive FIFO empty flag (bit 1, 2D7H) before read
–read dev. address	2D6H	
–read 1. databyte	2D6H	
–read 2. databyte	2D6H	

The value 21H in the first step results from the device address in the 7 MSBs and the R/W-bit (read=1) in the LSB. If the telegrams are longer, the software has to ensure that neither the Control-Data-FIFO nor the Read-FIFO can overflow.

To write data to this device:

–write 20H to	2D0H
–write 1. databyte to	2D2H
–write 2. databyte to	2D4H

The bus activity starts immediately after the first write to the Control-Data-FIFO. In the I²C mode the transmission can be synchronized by an artificial extension of the Low phase of the clock line. Transmission is not continued until the state of the clock line is High once again. Thus a slave (software slaves!) can adjust the transmission rate to its own abilities.

The I²C/IM bus interface is a pure Master system, Multimaster busses are not realizable.
The ident, clock and data terminal pins have open-drain outputs with weak pull-up transistors.

UNIT = Bit-Period

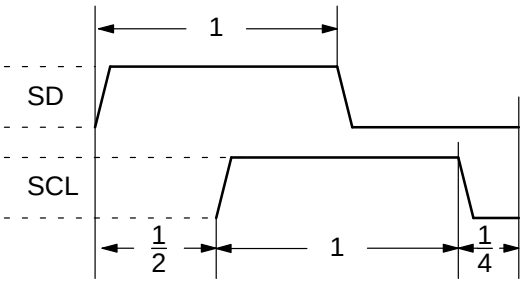


Fig. 2–20: Start condition I²C bus

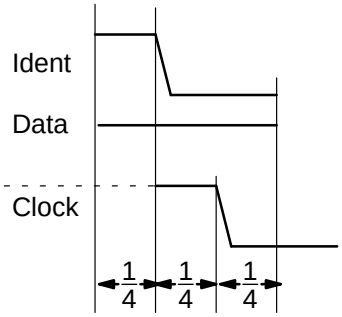


Fig. 2–23: IM bus start condition

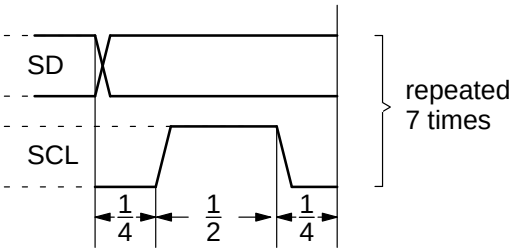


Fig. 2–21: Single bit on I²C bus

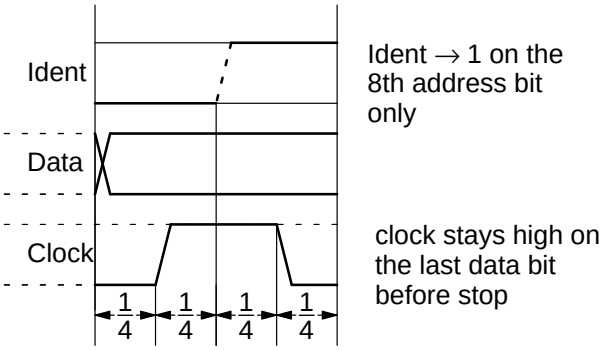


Fig. 2–24: Single bit on IM bus

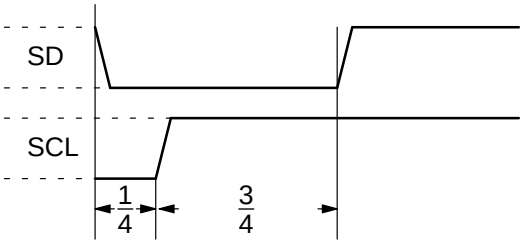


Fig. 2–22: Stop condition I²C bus

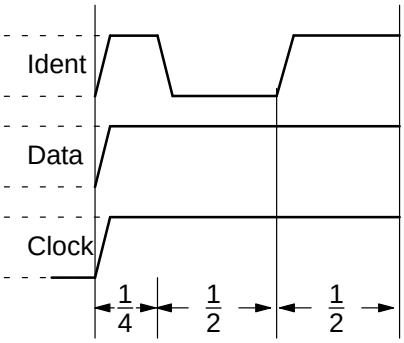


Fig. 2–25: Stop condition IM bus

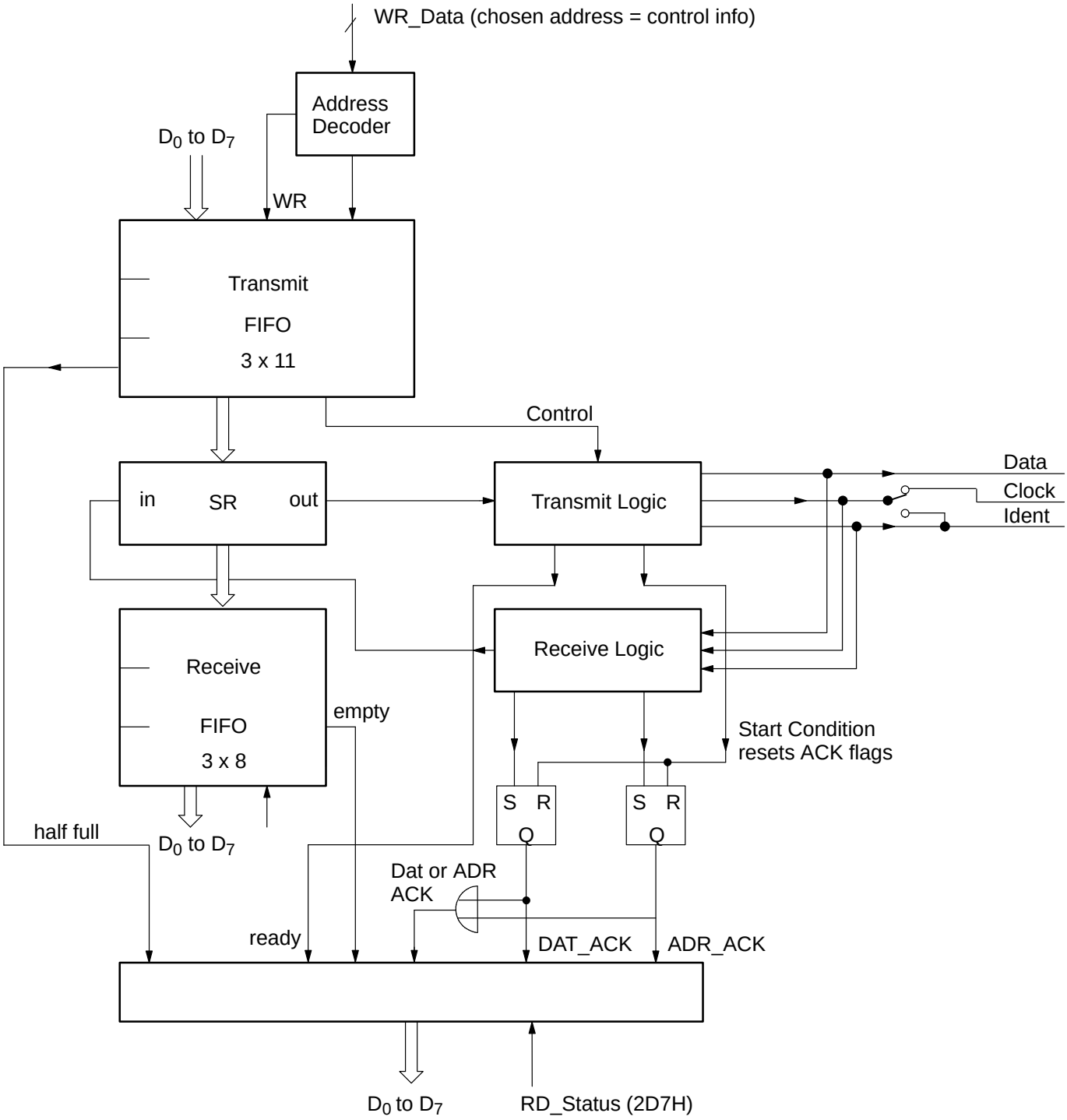


Fig. 2–26: I²C/IM bus interface

8.7. Pin Connections and Short Descriptions

DA = IM bus data line of external devices
ID = IM bus ident line of external devices
CL = IM bus clock line of external devices
SDA = I^2C bus data line of external devices
SCL = I^2C bus clock line of external devices
X = obligatory; connections depend on application

Pin No. 68–pin PLCC	Pin No. 64–pin SDIP	Con-nection	I: Input O: Output	Pin Name	Short Description
1	16	+5V	I	V_{SUP}	Supply Voltage
2	15	GND	I	GND	Ground
3	14	Xtal	I/O	X2	Crystal Connector 2
4	13	Xtal	I	X1	Crystal Connector 1
5	12	+3V to +5V	I	$V_{standby}$	Standby Supply Voltage
6	11	X	I/O	$\overline{RES}$	Reset input / Reset output
7	10	DA/SDA	I/O	IM1_DAT/I^2C_DAT	IM bus 1 Data line/ I^2C Bus Data line
8	9	ID/SCL	O	IM1_ID/I^2C_CLK2	IM bus 1 Ident line/ I^2C Bus Clock line
9	8	CL/SCL	O	IM1_CLK/I^2C_CLK1	IM bus 1 Clock line/ I^2C Bus Clock line
10	–	DA	I/O	IM2_DAT	IM bus 2 Data line
11	–	ID	O	IM2_ID	IM bus 2 Ident line
12	–	CL	O	IM2_CLK	IM bus 2 Clock line
13	7	X	I/O	TIMER1	Timer 1 input/output
14	6	X	I/O	TIMER2	Timer 2 input/output
15	5	X	I/O	TIMER3	Timer 3 input/output
16	4	external infrared receiver	I	IR	Infrared Signal Input
17	3	X	I/O (O)	P4 (R/$\overline{W}$)	Port 4, Bit 0 (CPU read/write)
18	2	X	I/O (I/O)	P10 (D0)	Port 1, Bit 0 (CPU data bus bit 0)
19	1	X	I/O (I/O)	P11 (D1)	Port 1, Bit 1 (CPU data bus bit 1)
20	64	X	I/O (I/O)	P12 (D2)	Port 1, Bit 2 (CPU data bus bit 2)
21	63	X	I/O (I/O)	P13 (D3)	Port 1, Bit 3 (CPU data bus bit 3)
22	62	X	I/O (I/O)	P14 (D4)	Port 1, Bit 4 (CPU data bus bit 4)

Pin No. 68–pin PLCC	Pin No. 64–pin SDIP	Con-nection	I: Input O: Output	Pin Name	Short Description
23	61	X	I/O (I/O)	P15 (D5)	Port 1, Bit 5 (CPU data bus bit 5)
24	60	X	I/O (I/O)	P16 (D6)	Port 1, Bit 6 (CPU data bus bit 6)
25	59	X	I/O (I/O)	P17 (D7)	Port 1, Bit 7 (CPU data bus bit 7)
26	58	X	I/O (O)	P20 (A0)	Port 2, Bit 0 (CPU address bit 0)
27	57	X	I/O (O)	P21 (A1)	Port 2, Bit 1 (CPU address bit 1)
28	56	X	I/O (O)	P22 (A2)	Port 2, Bit 2 (CPU address bit 2)
29	55	X	I/O (O)	P23 (A3)	Port 2, Bit 3 (CPU address bit 3)
30	54	X	I/O (O)	P24 (A4)	Port 2, Bit 4 (CPU address bit 4)
31	53	X	I/O (O)	P25 (A5)	Port 2, Bit 5 (CPU address bit 5)
32	52	X	I/O (O)	P26 (A6)	Port 2, Bit 6 (CPU address bit 6)
33	51	X	I/O (O)	P27 (A7)	Port 2, Bit 7 (CPU address bit 7)
34	50	X	I/O (O)	P30 (A8)	Port 3, Bit 0 (CPU address bit 8)
35	49	X	I/O (O)	P31 (A9)	Port 3, Bit 1 (CPU address bit 9)
36	48	X	I/O (O)	P32 (A10)	Port 3, Bit 2 (CPU address bit 10)
37	47	X	I/O (O)	P33 (A11)	Port 3, Bit 3 (CPU address bit 11)
38	46	X	I/O (O)	P34 (A12)	Port 3, Bit 4 (CPU address bit 12)
39	45	X	I/O (O)	P35 (A13)	Port 3, Bit 5 (CPU address bit 13)
40	44	X	I/O (O)	P36 (A14)	Port 3, Bit 6 (CPU address bit 14)
41	43	X	I/O (O)	P37 (A15)	Port 3, Bit 7 (CPU address bit 15)
42	42	X	I/O (O)	P50 ($\overline{RD}$ Port 1)	Port 5, Bit 0 (CCU read Port 1)
43	41	X	I/O (O)	P51 ($\overline{WR}$ Port 1)	Port 5, Bit 1 (CCU write Port 1)
44	40	X	I/O (O)	P52 ($\overline{RD}$ Port 2)	Port 5, Bit 2 (CCU read Port 2)
45	39	X	I/O (O)	P53 ($\overline{WR}$ Port 2)	Port 5, Bit 3 (CCU write Port 2)
46	38	X	I/O (O)	P54 ($\overline{RD}$ Port 3)	Port 5, Bit 4 (CCU read Port 3)
47	37	X	I/O (O)	P55 ($\overline{WR}$ Port 3)	Port 5, Bit 5 (CCU write Port 3)
48	36	X	I/O (O)	P70 (Memory Bank Address 0)	Port 7, Bit 0 (Memory Bank Address 0)
49	35	X	I/O (O)	P71 (Memory Bank Address 1)	Port 7, Bit 1 (Memory Bank Address 1)
50	34	X	I/O (O)	P72 (Memory Bank Address 2)	Port 7, Bit 2 (Memory Bank Address 2)

Pin No. 68–pin PLCC	Pin No. 64–pin SDIP	Con-nection	I: Input O: Output	Pin Name	Short Description
51	33	X	I/O (O)	P73 (Memory Bank Address 3)	Port 7, Bit 3 (Memory Bank Address 3)
52	32	X	I/O (O)	P74 (Memory Bank Address 4)	Port 7, Bit 4 (Memory Bank Address 4)
53	31	X	I/O (O)	P75 (Memory Bank Address 5)	Port 7, Bit 5 (Memory Bank Address 5)
54	30	X	I/O (O)	P76 ($\overline{R}$/W)	Port 7, Bit 6 (CPU read/write signal)
55	29	X	I/O (O)	P77 (Power-Down Control)	Port 7, Bit 7 (Power-Down Control)
56	28	X	I/O	P80	Port 8, Bit 0
57	27	X	I/O /I	P81/INT	Port 8, Bit 1 / Interrupt input
58	26	X	I/O	P82	Port 8, Bit 2
59	–	X	I/O	P83	Port 8, Bit 3
60	25	X	I/O /I	P87/INT	Port 8, Bit 7/Interrupt input
61	24	X	I/O	P60	Port 6, Bit 0
62	23	X	I/O	P61	Port 6, Bit 1
63	22	X	I/O	P62	Port 6, Bit 2
64	21	X	I/O	P63	Port 6, Bit 3
65	20	X	I/O	P64	Port 6, Bit 4
66	19	X	I/O	P65	Port 6, Bit 5
67	18	X	I/O	P66	Port 6, Bit 6
68	17	X	I/O	P67	Port 6, Bit 7

8.7.1. DC Parameters I²C Bus Master Interface

The input and output parameters of the I²C bus interface (Clock and Data) are designed according to the ITT specification for Port and IM bus pins (the interface can also be operated as IM bus interface). The differences are:

Symbol	Meaning	ITT	I²C Specification
U_{IL}	Input Low Voltage	max. 1 V	max. 1.5 V
U_{IH}	Input High Voltage	min. 2.8 V	min. 3 V
U_{OL}	Output Low Voltage	0.4 V / 2 mA	0.4 V / 3 mA

The ITT parameters are equivalent to software I²C bus solutions using Port-lines for the bus. In applications with series resistors in the clock or data line these differences may become important.

8.8. List of Registers that Differ from CCU 3000, CCU 3001

The IM1 Registers of CCU3000 (Addr. from 0210H to 021BH) are no longer available.

02D0H	I²C Start Cycle without Generation of ACK (ACK = 1)		
Bit	**Reset**	**Read**	**Write**
7 to 0	x	x	I²C-Start-Data

02D1H	I²C Start Cycle with Generation of ACK (ACK=0)		
Bit	**Reset**	**Read**	**Write**
7 to 0	x	x	I²C-Start-Data

02D2H	I²C Resume Cycle without Generation of ACK (ACK = 1)		
Bit	**Reset**	**Read**	**Write**
7 to 0	x	x	I²C-Resume-Data *(set this byte to $FF for a read access)*

02D3H	I²C Resume Cycle with Generation of ACK (ACK=0)		
Bit	**Reset**	**Read**	**Write**
7 to 0	x	x	I²C-Resume-Data *(set this byte to $FF for a read access)*

02D4H	I²C Termination Cycle without Generation of ACK (ACK =1)		
Bit	**Reset**	**Read**	**Write**
7 to 0	x	x	I²C-Terminate-Data *(set this byte to $FF for a read access)*

02D5H	I²C Termination Cycle with Generation of ACK (ACK=0)		
Bit	**Reset**	**Read**	**Write**
7 to 0	x	x	I²C-Terminate-Data *(set this byte to $FF for a read access)*

02D6H	I²C/IM Bus Data from Receive–FIFO		
Bit	**Reset**	**Read**	**Write**
7 to 0	x	Received data	x

02D7H	I²C/IM Bus Status		
Bit	**Reset**	**Read**	**Write**
7	0	x	x
6	0	I²C OR'D ACK	x
5	0	I²C ADDR–ACK	x
4	0	I²C Data–ACK	x
3	0	Bus busy	x
2	0	WR FIFO half full	x
1	0	RD FIFO empty	x
0	0	x	x

02D8H	IM Bus Start Cycle		
Bit	**Reset**	**Read**	**Write**
7 to 0	x	x	IM bus start–(address)–data

02D9H	IM Bus Resume Cycle		
Bit	**Reset**	**Read**	**Write**
7 to 0	x	x	IM bus resume data

02DAH	IM Bus Termination Cycle		
Bit	**Reset**	**Read**	**Write**
7 to 0	x	x	IM bus terminal data

02DBH	I²C/IM Bus Prescaler		
Bit	**Reset**	**Read**	**Write**
7	1	x	$\overline{\text{select I}^2\text{C\_CLK2 on IM1\_ID}}$ / select I²C_CLK1 on IM1_CLK
6 to 0	0	x	$f_{IMI2C} = f_{OSC} / (4*n)$ for n>1 $f_{IMI2C} = f_{OSC} / 6$ for n=1 f_{IMI2C} not running for n=0

9. Data Sheet History

1. Data Sheet “CCU 3000, CCU 3000-I, CCU 3001”,
Feb. 14, 1995, 6251-367-1DS:
First release of the data sheet.

ITT Semiconductors Group
World Headquarters
INTERMETALL
Hans-Bunte-Strasse 19
D-79108 Freiburg (Germany)
P.O. Box 840
D-79008 Freiburg (Germany)
Tel. +49-761-517-0
Fax +49-761-517-174

Printed in Germany
by Simon Druck GmbH & Co., Freiburg (02/95)
Order No. 6251-367-1DS

End of Data Sheet

Back to Summary

Back to Data Sheets ICs

Semiconductors