

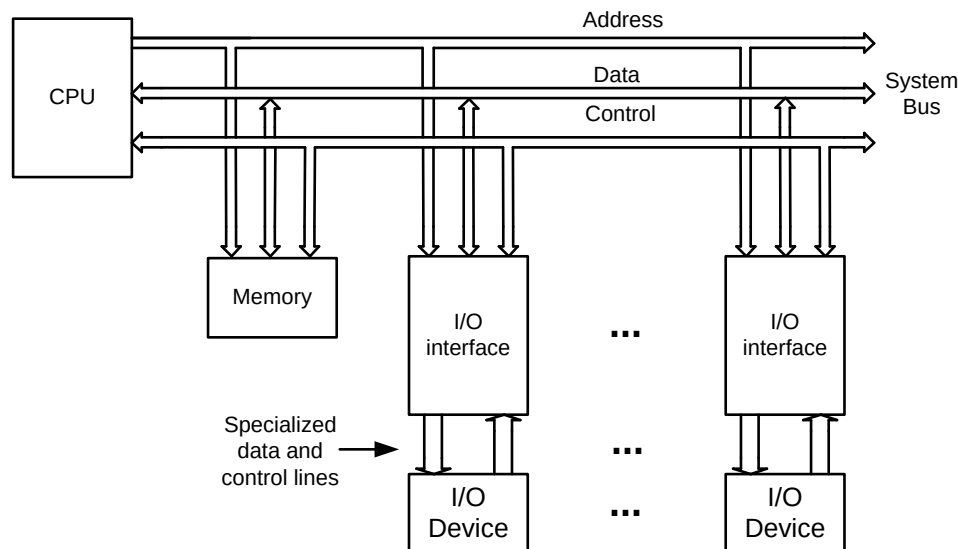
Chapter 7 Input/Output System

7.1 Terminology

- ***I/O bus***: contains data, address, and control lines
- ***I/O (or peripheral) device***: performs some function for computer
- ***I/O (or device) interface***: controls the operation of an I/O device according to commands from processor; it also converts data into whatever format is required by the device and vice versa.
- ***I/O port (or I/O register)***: is a part of a device interface, a group of bits accessed by the processor during I/O operations.
- ***Parallel I/O***: the data exchange between peripheral device and processor is in parallel format, typically of 8-bit or 16-bit wide; often used to connect local devices
- ***Serial I/O***: the data exchange between peripheral device and processor is in serial format, one bit at time, often used to connect remote devices
- ***Handshaking***: a protocol for establishing a valid data transfer between peripheral device and processor
- ***Polling***: a technique to check the readiness of data or device
- ***Interrupt***: a technique to alert the processor for servicing

7.2 Memory-mapped I/O

- shares the same address and data buses with memory system
- input and output operations are treated as a memory read/write operation



Advantages:

- All memory reference instruction may be used to manipulate I/O ports
- The number of available I/O port address is virtually unlimited
- The hardware bus structure is simplified

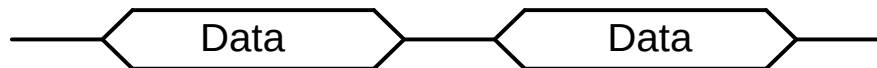
Disadvantages:

- Part of the memory address is used up
- Interface may need a complicated decoder circuit to recognize a long address

7.3 Parallel Data I/O

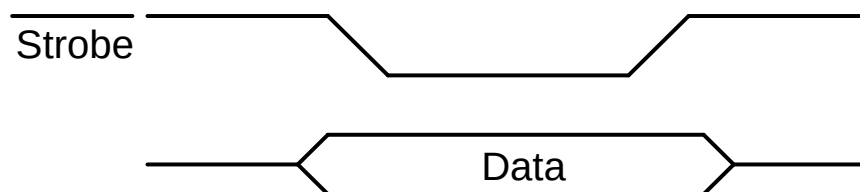
7.3.1 Simple Input and Output

CPU can simply read digital data presented to the input port (e.g. switches or keyboard) and place data to the output port (e.g. turn on/off LEDs for display) at any time.



7.3.2 Simple Strobe I/O

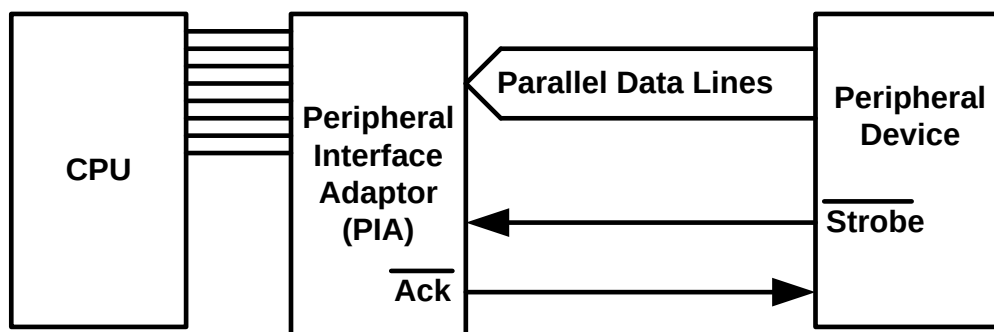
If the input data is only appeared for a limited time, CPU must read in the data when it is still valid. Usually a **Strobe** signal is used to inform CPU when data is ready.



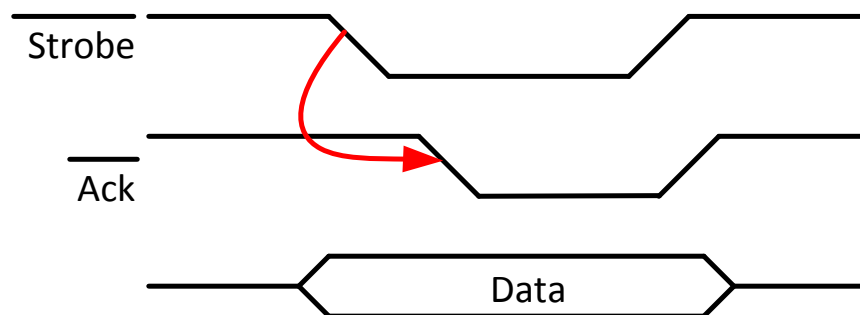
Strobe signal may be connected to extra input line or to the interrupt line. The CPU must complete the read operation when the **Strobe** signal is active. The input device does not need to check whether if the CPU had read the data or not.

7.3.2 Single-handshake I/O

To ensure successful data transfer, it is better to include handshaking signals between CPU and peripheral. This is generally achieved using a peripheral interface adaptor (PIA) which can isolate the peripheral device from the system buses. In addition the PIA can be programmed to automatically manage the handshake operation without the intervention of CPU. The following diagram shows the connection for an input system.



If the input device has a data ready for CPU, it will send a **Strobe** signal to alert CPU. Upon reading the data, CPU will inform the device by an **Ack** signal as shown below.

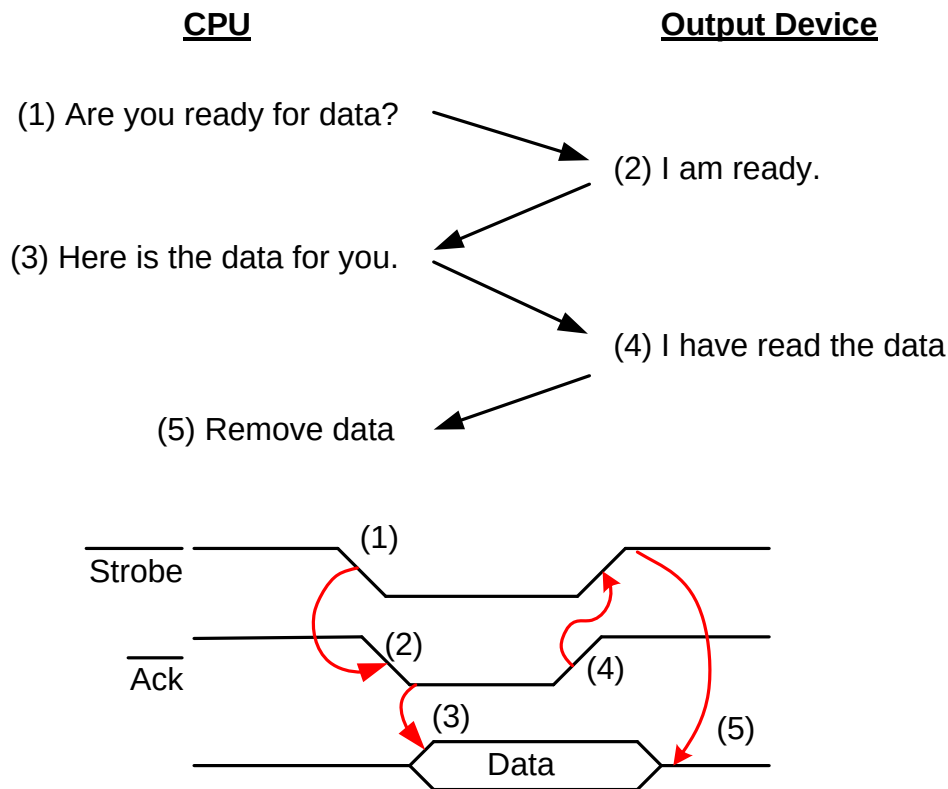


For output data from CPU, CPU will generate the **Strobe** signal and the device will return the **Ack** signal in order ensure that the device actually receive the data.

In single-handshake, sender is not required to verify the reception of the data on the receiving end.

7.3.3 Double-handshake I/O

Double handshake is used when a more rigorous coordination for data transfer is required. The hardware connection is same as that for the single handshake I/O system. The handshaking protocol is somewhat like the diagram shown in the following diagram.



The monitoring of the handshaking signal **Strobe** and **Ack** can be implemented using port pins. However, this requires CPU to monitor the status change on these pins. A more efficient method is to use these to generate interrupt signal to alert CPU for appropriate action.

7.4 I/O port

- I/O devices will not be directly connected to the system data bus since the data will only appear during the bus cycle and changed for the next bus cycle.
- The I/O device may not be able to capture the data if its operational speed is relatively slower than the CPU clock frequency.
- A latch is required to “hold” the data for the device to respond.
- The concept of “port” is adopted to act as an interface between CPU and I/O device. It is mainly used to latch the data (simple circuit).
- The hardware interfacing signals can be isolated from CPU to make circuit much simpler.

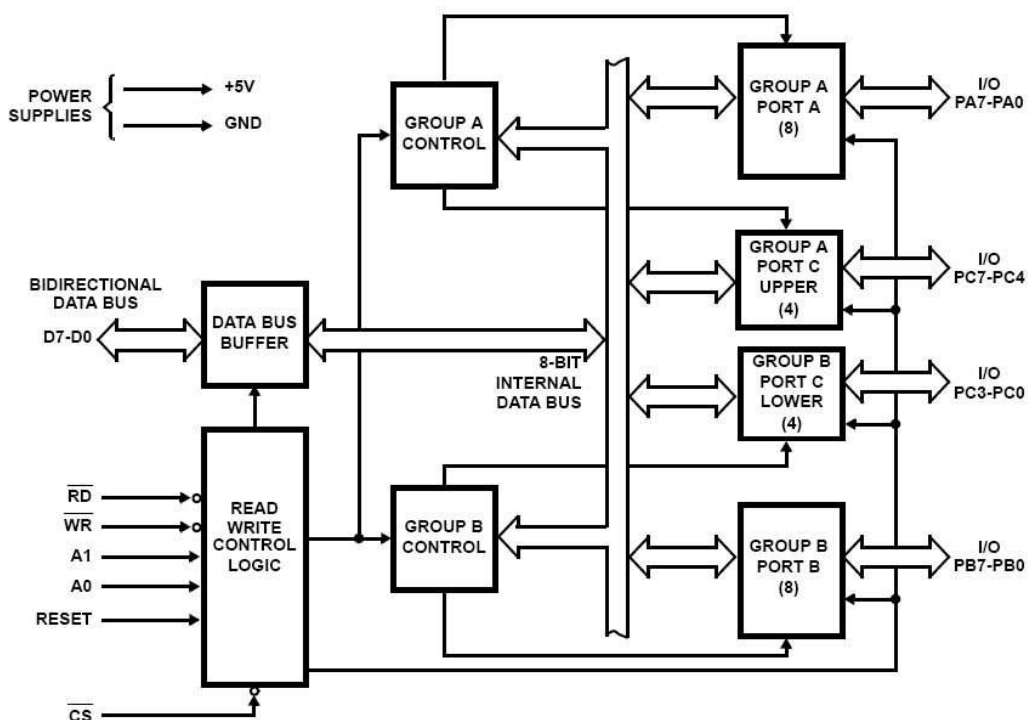
7.4.1 I/O port using Peripheral Interface Adaptor (PIA) – an example 8255A

For some devices, there may be many signals required to be connected to the system bus and causing a complicated hardware design, using a peripheral interface adaptor (PIA) can isolate the device from the system and resulting in a simpler hardware design.

8255A a general purpose Programmable Peripheral Interface Adaptor (PIA) device for interfacing I/O devices and microprocessor without integrated with I/O ports.

Features of 8255A:

- Three 8-bit I/O ports (ports A, B and C)
- Port C can be used as I/O port or handshake I/O port
- Port A can be operated for bidirectional handshake data transfer, i.e. data can be output or input on the same 8 lines.
- Interrupt request signal is available to interrupt CPU.
- Use Control register to configure the operation mode.
- Use Status register to monitor the status of the interface adaptor.
- Data registers for data transfer.



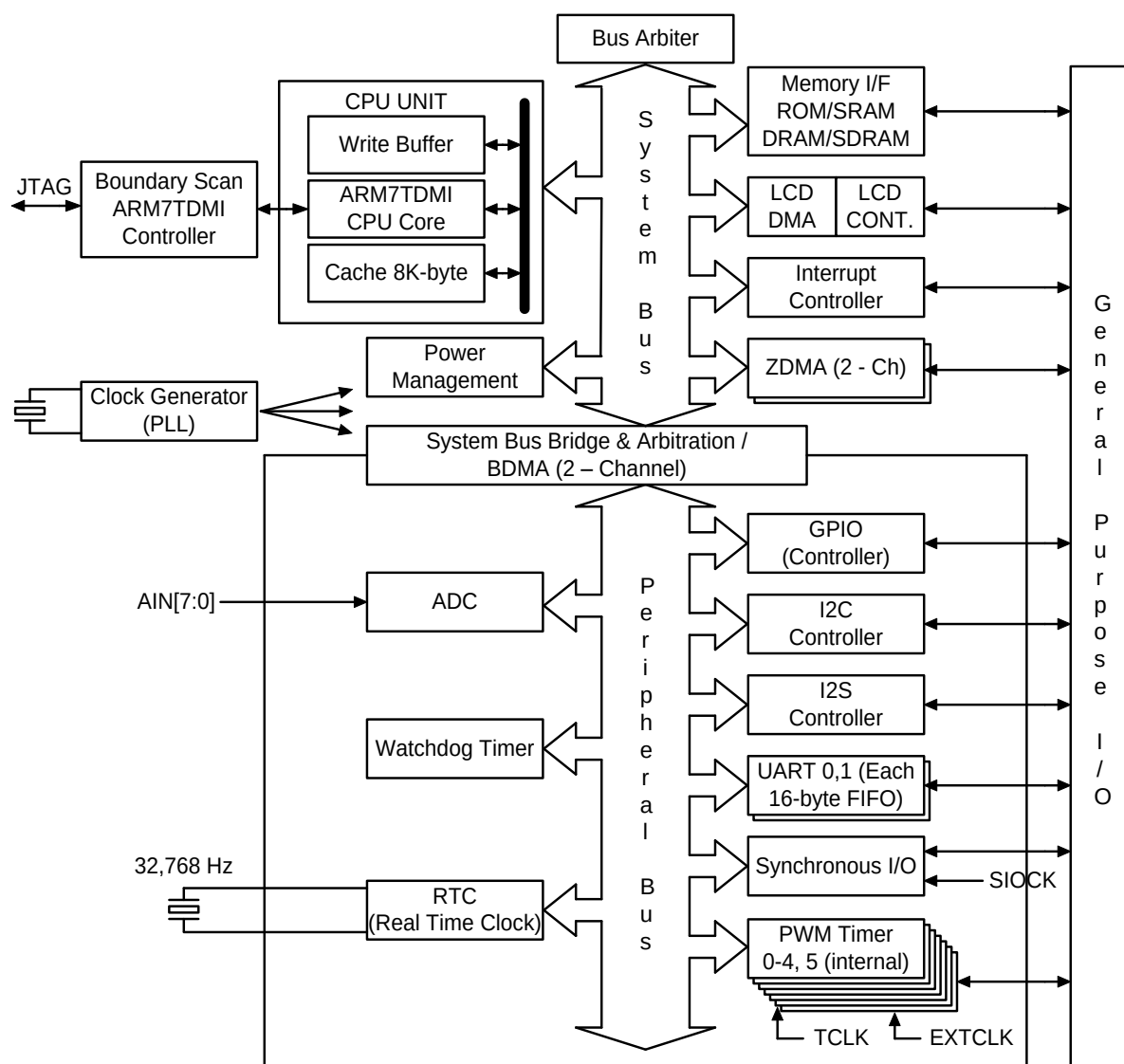
Functional Diagram of 8255A

7.5 I/O Ports of S3C44B0X

S3C44B0X has seven ports:

- One 10-bit output port. (Port A)
- One 11-bit output port. (Port B)
- One 16-bit input/output port. (Port C)
- Two 8-bit input/output ports. (Port D and G)
- Two 9-bit input/output ports. (Port E and F)

- Most port pins are multiplexed pins
- Pin function is programmable



System Bus: high speed

Peripheral Bus: slow speed

All I/O registers of peripherals are memory mapped.

7.5.1 Port Functions

Port A	Selectable Pin functions	
	Function 1	Function 2
PA9	output only	ADDR24
PA8	output only	ADDR23
PA7	output only	ADDR22
PA6	output only	ADDR21
PA5	output only	ADDR20
PA4	output only	ADDR19
PA3	output only	ADDR18
PA2	output only	ADDR17
PA1	output only	ADDR16
PA0	output only	ADDR0

Port B	Selectable Pin functions	
	Function 1	Function 2
PB10	output only	nGCS5
PB9	output only	nGCS4
PB8	output only	nGCS3
PB7	output only	nGCS2
PB6	output only	nGCS1
PB5	output only	nWBE3:nBE3:DQM3
PB4	output only	nWBE2:nBE:DQM2
PB3	output only	nSRAS:nCAS3
PB2	output only	nSCAS:nCAS2
PB1	output only	SCLK
PB0	output only	SCKE

Port C	Selectable Pin functions		
	Function 1	Function 2	Function 3
PC15	Input/output	<u>DATA31</u>	nCTS0
PC14	Input/output	<u>DATA30</u>	nRTS0
PC13	Input/output	<u>DATA29</u>	RxD1
PC12	Input/output	<u>DATA28</u>	TxD1
PC11	Input/output	<u>DATA27</u>	nCTS1
PC10	Input/output	<u>DATA26</u>	nRTS1
PC9	Input/output	<u>DATA25</u>	nXDREQ1
PC8	Input/output	<u>DATA24</u>	nXDACK1
PC7	Input/output	<u>DATA23</u>	VD4
PC6	Input/output	<u>DATA22</u>	VD5
PC5	Input/output	<u>DATA21</u>	VD6
PC4	Input/output	<u>DATA20</u>	VD7
PC3	Input/output	<u>DATA19</u>	IISCLK
PC2	Input/output	<u>DATA18</u>	IISD1
PC1	Input/output	<u>DATA17</u>	IISDO
PC0	Input/output	<u>DATA16</u>	IISLRCK

Port D	Selectable Pin functions	
	Function 1	Function 2
PD7	<u>Input/output</u>	VFRAME
PD6	<u>Input/output</u>	VM
PD5	<u>Input/output</u>	VLINE
PD4	<u>Input/output</u>	VCLK
PD3	<u>Input/output</u>	VD3
PD2	<u>Input/output</u>	VD2
PD1	<u>Input/output</u>	V1
PD0	<u>Input/output</u>	VD0

Port E	Selectable Pin functions		
	Function 1	Function 2	Function 3
PE8	<u>ENDIAN</u>	CODECLK	Input/output
PE7	<u>Input/output</u>	TOUT4	VD7
PE6	<u>Input/output</u>	TOUT3	VD6
PE5	<u>Input/output</u>	TOUT2	TCLK
PE4	<u>Input/output</u>	TOUT1	TCLK
PE3	<u>Input/output</u>	TOUT0	-
PE2	<u>Input/output</u>	RxD0	-
PE1	<u>Input/output</u>	TxD0	-
PE0	<u>Input/output</u>	FpIlo	Fout

Port F	Selectable Pin functions			
	Function 1	Function 2	Function 3	Function 4
PF8	<u>ENDIAN</u>	nCTS1	SIOCK	IISCLK
PF7	<u>Input/output</u>	RxD1	SIORxD	IISDI
PF6	<u>Input/output</u>	TxD1	SIORDY	IISDO
PF5	<u>Input/output</u>	nRTS1	SIOTxD	IISLRCK
PF4	<u>Input/output</u>	nXBREQ	nXBREQ0	-
PF3	<u>Input/output</u>	nXBACK	nXDACK0	-
PF2	<u>Input/output</u>	nWAIT	-	-
PF1	<u>Input/output</u>	IICSDA	-	-
PF0	<u>Input/output</u>	IICSCL	-	-

Port G	Selectable Pin functions		
	Function 1	Function 2	Function 3
PG7	<u>Input/output</u>	IISLRCK	EINT7
PG6	<u>Input/output</u>	IISDI	EINT6
PG5	<u>Input/output</u>	IISDO	EINT5
PG4	<u>Input/output</u>	IISCLK	EINT4
PG3	<u>Input/output</u>	nRTS0	EINT3
PG2	<u>Input/output</u>	nCTS0	EINT2
PG1	<u>Input/output</u>	VD5	EINT1
PG0	<u>Input/output</u>	VD4	EINT0

7.5.2 Port Registers

Port Configuration Register (PCON_i, $i=A,B,C,D,E,F,G$)

In S3C44B0X, the functions for each pin should be selected. The PCON_n (port control register) determines which function is used for each pin. If PG0 - PG7 are used for the wakeup signal in power down mode, these ports must be configured in interrupt mode.

Port Data Register (PDAT_i, $i= A,B,C,D,E,F,G$)

If these ports are configured as output ports, data can be written to the corresponding bit of PDAT_n. If Ports are configured as input ports, the data can be read from the corresponding bit of PDAT_n.

Port Pull-up Register (PUP_i, $i= C,D,E,F,G$)

The port pull-up resistor controls the pull-up resistor enable/disable of each port group. When the corresponding bit is 0, the pull-up resistor of the pin is enabled (all pull-up resistor are enabled upon reset). When 1, the pull-up resistor is disabled.

External Interrupt Control Register

The 8 external interrupts are requested by various signaling methods. The EXTINT register configures the signaling method among the low level trigger, high level trigger, falling edge trigger, rising edge trigger, and both edge triggers for the external interrupt request.

Because each external interrupt pin has a digital filter, the interrupt controller can recognize the request signal longer than 3 clocks.

I/O PORT CONTROL REGISTER**PORT A CONTROL REGISTERS (PCONA, PDATA, PUPA)**

Port A control registers are shown in Table 8-2:

Register	Address	R/W	Description	Reset Value
PCONA	0x01D20000	R/W	Configures the pins of port A	0x3FF
PDATA	0x01D20004	R/W	The data register for port A	Undef.

NOTE: The Reset value of 0x3ff for PCONA sets PA0-PA9 as ADDRESS pins

Table 8-2. Port of Group A Control Registers (PCONA, PDATA)

PCONA	Bit	Description	
PA9	[9]	0 = Output	1 = ADDR24
PA8	[8]	0 = Output	1 = ADDR23
PA7	[7]	0 = Output	1 = ADDR22
PA6	[6]	0 = Output	1 = ADDR21
PA5	[5]	0 = Output	1 = ADDR20
PA4	[4]	0 = Output	1 = ADDR19
PA3	[3]	0 = Output	1 = ADDR18
PA2	[2]	0 = Output	1 = ADDR17
PA1	[1]	0 = Output	1 = ADDR16
PA0	[0]	0 = Output	1 = ADDR15

PDATA	Bit	Description
PA[9:0]	[9:0]	When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as a functional pin, an undefined value will be read.

PORT B CONTROL REGISTERS (PCONB, PDATB)

Port B control registers are shown in Table 8-3:

Register	Address	R/W	Description	Reset Value
PCONB	0x01D20008	R/W	Configures the pins of port B	0x7FF
PDATB	0x01D2000C	R/W	The data register for port B	Undef.

Table 8-3. Port of Group B Control Registers (PCONB,PDATB)

PCONB	Bit	Description	
PB10	[10]	0 = Output	1 = nGCS5
PB9	[9]	0 = Output	1 = nGCS4
PB8	[8]	0 = Output	1 = nGCS3
PB7	[7]	0 = Output	1 = nGCS2
PB6	[6]	0 = Output	1 = nGCS1
PB5	[5]	0 = Output	1 = nWBE3/nBE3/DQM3
PB4	[4]	0 = Output	1 = nWBE2/nBE2/DQM2
PB3	[3]	0 = Output	1 = nSCAS:nCAS2
PB2	[2]	0 = Output	1 = nSCAS:nCAS2
PB1	[1]	0 = Output	1 = SCLK
PB0	[0]	0 = Output	1 = SCKE

PDATB	Bit	Description
PB[10:0]	[10:0]	When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as a functional pin, an undefined value will be read.

PORT C CONTROL REGISTERS (PCONC, PDATC, PUPC)

Port C control registers are shown in Table 8-4:

Register	Address	R/W	Description	Reset Value
PCONC	0x01D20010	R/W	Configures the pins of port C	0xaaaaaaaa
PDATC	0x01D20014	R/W	The data register for port C	Undef.
PUPC	0x01D20018	R/W	Pull-up disable register for port C	0x0

NOTE: The Reset value of 0xaaaaaaaa for PCONC sets PC0-PA15 as DATA pins

Table 8-4. Port of Group C Control Registers (PCONC,PDATC,PUPC)

PCONC	Bit	Description			
PC15	[31:30]	00 = Input	01 = Output	10 = DATA31	11 = nCTS0
PC14	[29:28]	00 = Input	01 = Output	10 = DATA30	11 = nRTS0
PC13	[27:26]	00 = Input	01 = Output	10 = DATA29	11 = Rx D1
PC12	[25:24]	00 = Input	01 = Output	10 = DATA28	11 = Tx D1
PC11	[23:22]	00 = Input	01 = Output	10 = DATA27	11 = nCTS1
PC10	[21:20]	00 = Input	01 = Output	10 = DATA26	11 = nRTS1
PC9	[19:18]	00 = Input	01 = Output	10 = DATA25	11 = nXDREQ1
PC8	[17:16]	00 = Input	01 = Output	10 = DATA24	11 = nXACK1
PC7	[15:14]	00 = Input	01 = Output	10 = DATA23	11 = VD4
PC6	[13:12]	00 = Input	01 = Output	10 = DATA22	11 = VD5
PC5	[11:10]	00 = Input	01 = Output	10 = DATA21	11 = VD6
PC4	[9:8]	00 = Input	01 = Output	10 = DATA20	11 = VD7
PC3	[7:6]	00 = Input	01 = Output	10 = DATA19	11 = IISCLK
PC2	[5:4]	00 = Input	01 = Output	10 = DATA18	11 = IISDI
PC1	[3:2]	00 = Input	01 = Output	10 = DATA17	11 = IISDO
PC0	[1:0]	00 = Input	01 = Output	10 = DATA16	11 = IISLRCK

PDATC	Bit	Description
PC[15:0]	[15:0]	When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as a functional pin, an undefined value will be read.

PUPC	Bit	Description
PC[15:0]	[15:0]	0: the pull up resistor attached to the corresponding port pin is enabled. 1: the pull up resistor is disabled.

PORT D CONTROL REGISTERS (PCOND, PDATD, PUPD)

Port D control registers are shown in Table 8-5.

Register	Address	R/W	Description	Reset Value
PCOND	0x01D2001C	R/W	Configures the pins of port D	0x0000
PDATD	0x01D20020	R/W	The data register for port D	Undef.
PUPD	0x01D20024	R/W	Pull-up disable register for port D	0x0

Port of Group D Control Registers (PCOND, PDATD, PUPD)

PCOND	Bit	Description			
PD7	[15:14]	00 = Input	01 = Output	10 = VFRAME	11 = Reserved
PD6	[13:12]	00 = Input	01 = Output	10 = VM	11 = Reserved
PD5	[11:10]	00 = Input	01 = Output	10 = VLINE	11 = Reserved
PD4	[9:8]	00 = Input	01 = Output	10 = VCLK	11 = Reserved
PD3	[7:6]	00 = Input	01 = Output	10 = VD3	11 = Reserved
PD2	[5:4]	00 = Input	01 = Output	10 = VD2	11 = Reserved
PD1	[3:2]	00 = Input	01 = Output	10 = VD1	11 = Reserved
PD0	[1:0]	00 = Input	01 = Output	10 = VD0	11 = Reserved

PDATD	Bit	Description
PD[7:0]	[7:0]	When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as a functional pin, an undefined value will be read.

PUPD	Bit	Description
PD[7:0]	[7:0]	0: the pull up resistor attached to the corresponding port pin is enabled. 1: the pull up resistor is disabled.

PORT E CONTROL REGISTERS (PCONE, PDATE)

Port E control registers are shown in Table 8-6:

Register	Address	R/W	Description	Reset Value
PCONE	0x01D20028	R/W	Configures the pins of port E	0x00
PDATE	0x01D2002C	R/W	The data register for port E	Undef.
PUPE	0x01D20030	R/W	Pull-up disable register for port E	0x0

Table 8-6. Port of Group E Control Registers (PCONE, PDATE)

PCONE	Bit	Description			
PE8	[17:16]	00 = Reserved (ENDIAN) 01 = Output 10 = CODECLK 11 = Reserved PE8 can be used as ENDIAN only during the reset cycle.			
PE7	[15:14]	00 = Input	01 = Output	10 = TOUT4	11 = VD7
PE6	[13:12]	00 = Input	01 = Output	10 = TOUT3	11 = VD6
PE5	[11:10]	00 = Input	01 = Output	10 = TOUT2	11 = TCLK in
PE4	[9:8]	00 = Input	01 = Output	10 = TOUT1	11 = TCLK in
PE3	[7:6]	00 = Input	01 = Output	10 = TOUT0	11 = Reserved
PE2	[5:4]	00 = Input	01 = Output	10 = RxD0	11 = Reserved
PE1	[3:2]	00 = Input	01 = Output	10 = TxD0	11 = Reserved
PE0	[1:0]	00 = Input	01 = Output	10 = FpI/O out	11 = Fout out

PDATE	Bit	Description
PE[8:0]	[8:0]	When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as a functional pin, the undefined value will be read.

PUPE	Bit	Description
PE[7:0]	[7:0]	0: the pull up resistor attached to the corresponding port pin is enabled. 1: the pull up resistor is disabled. PE8 do not have programmable pull-up resistor.

PORT F CONTROL REGISTERS (PCONF, PDATAF, PUPF)

Port F control registers are shown in Table 8-7 below:

Register	Address	R/W	Description	Reset Value
PCONF	0x01D20034	R/W	Configures the pins of port F	0x0000
PDATAF	0x01D20038	R/W	The data register for port F	Undef.
PUPF	0x01D2003C	R/W	Pull-up disable register for port F	0x0

Table 8-7. Port of Group F Control Registers (PCONF, PDATAF, PUPF)

PCONF	Bit	Description			
PF8	[21:19]	000 = Input 011 = SIOCLK	001 = Output 100 = IISCLK	010 = nCTS1 Others = Reserved	
PF7	[18:16]	000 = Input 011 = SIORxD	001 = Output 100 = IISD1	010 = Rx D1 Others = Reserved	
PF6	[15:13]	000 = Input 011 = SIORDY	001 = Output 100 = IISD0	010 = Tx D1 Others = Reserved	
PF5	[12:10]	000 = Input 011 = SIOTxD	001 = Output 100 = IISLRCK	010 = nRTS1 Others = Reserved	
PF4	[9:8]	00 = Input	01 = Output	10 = nXBREQ	11 = nXDREQ0
PF3	[7:6]	00 = Input	01 = Output	10 = nXBACK	11 = nXDACK0
PF2	[5:4]	00 = Input	01 = Output	10 = nWAIT	11 = Reserved
PF1	[3:2]	00 = Input	01 = Output	10 = IICSDA	11 = Reserved
PF0	[1:0]	00 = Input	01 = Output	10 = IIC SCL	11 = Reserved

PDATAF	Bit	Description
PF[8:0]	[8:0]	When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as a functional pin, an undefined value will be read.

PUPF	Bit	Description
PF[8:0]	[8:0]	0: the pull up resistor attached to the corresponding port pin is enabled. 1: the pull up resistor is disabled.

PORT G CONTROL REGISTERS (PCONG, PDATG, PUPG)

Port G control registers are shown in Table 8-8:

If PG0 - PG7 are to be used for wake-up signals in power down mode, the ports will be set in the interrupt mode.

Register	Address	R/W	Description	Reset Value
PCONG	0x01D20040	R/W	Configures the pins of port G	0x0
PDATG	0x01D20044	R/W	The data register for port G	Undef.
PUPG	0x01D20048	R/W	Pull-up disable register for port G	0x0

Table 8-8. Port of Group G Control Registers (PCONG, PDATG, PUPG)

PCONG	Bit	Description			
PG7	[15:14]	00 = Input	01 = Output	10 = IISLRCK	11 = EINT7
PG6	[13:12]	00 = Input	01 = Output	10 = IISDO	11 = EINT6
PG5	[11:10]	00 = Input	01 = Output	10 = IISDI	11 = EINT5
PG4	[9:8]	00 = Input	01 = Output	10 = IISCLK	11 = EINT4
PG3	[7:6]	00 = Input	01 = Output	10 = nRTS0	11 = EINT3
PG2	[5:4]	00 = Input	01 = Output	10 = nCTS0	11 = EINT2
PG1	[3:2]	00 = Input	01 = Output	10 = VD5	11 = EINT1
PG0	[1:0]	00 = Input	01 = Output	10 = VD4	11 = EINT0

PDATG	Bit	Description
PG[7:0]	[7:0]	When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as a functional pin, an undefined value will be read.

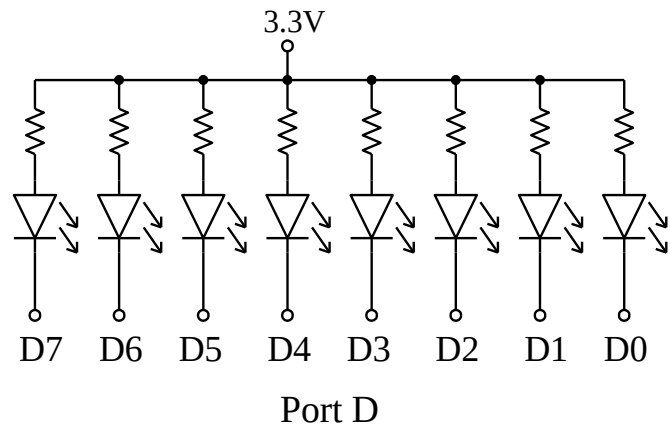
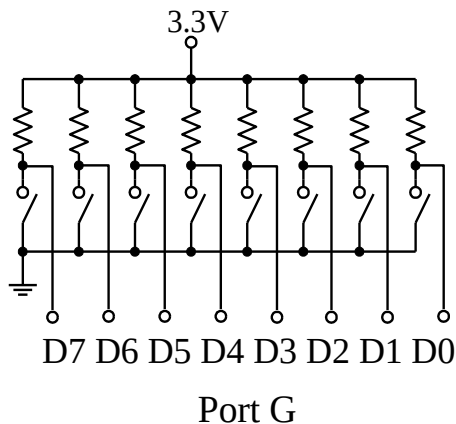
PUPG	Bit	Description
PG[7:0]	[7:0]	0: the pull up resistor attached to the corresponding port pin is enabled. 1: the pull up resistor is disabled.

7.5.3 Port Setting

Determine the ports from the hardware I/O connections and then:

- (i) Set $PCONG_i$ to select for function
- (ii) Set PUP_i to enable or disable the port pull-up resistors
- (iii) Read from or write to the port register $PDAT_i$.

Example as GPIO:



Program:

```
.equ  PconG, 0x01d20040 /* Port G Config. Port */
.equ  PdatG, 0x01d20044 /* Port G Data Port */
.equ  PconD, 0x01d2001c /* Port D Config. Port */
.equ  PdatD, 0x01d20020 /* Port D Data Port */

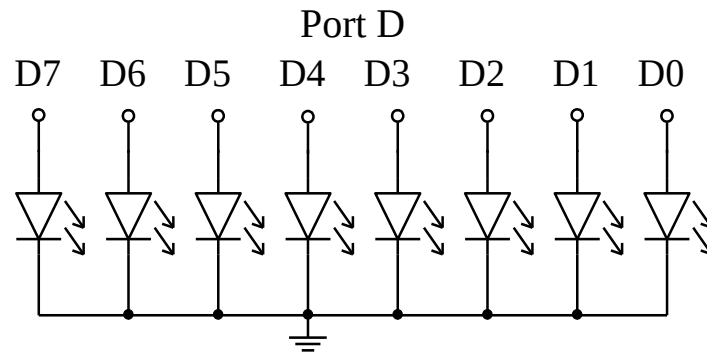
.global _start
.text
_start:
    LDR    R0, =0x5555 /* Setup the config. value to Port D */
    LDR    R1, =0x0000 /* Setup the config. value to Port G */
    LDR    R2, =PconD   /* R2 point to Port D config. register */
    LDR    R3, =PconG   /* R3 point to Port G config. register */
    STRH   R0, [R2]     /* Set Port D as O/P: PConD is 16 bits */
    STRH   R1, [R3]     /* Set Port G as I/P: PConG is 16 bits */
    LDR    R2, =PdatD   /* R2 point to PdatD - 8 bits */
    LDR    R3, =PdatG   /* R2 point to PdatG - 8 bits */

loop:   LDR    R4, [R3] /* Read the dil switch */
        STR    R4, [R2] /* O/P the data to LED */
        B      loop

.end
```

Note that the above circuit is for outputting a logic '0' to turn on the LED.

If a logic '1' is required to turn on the LED, the circuit will be as follow:

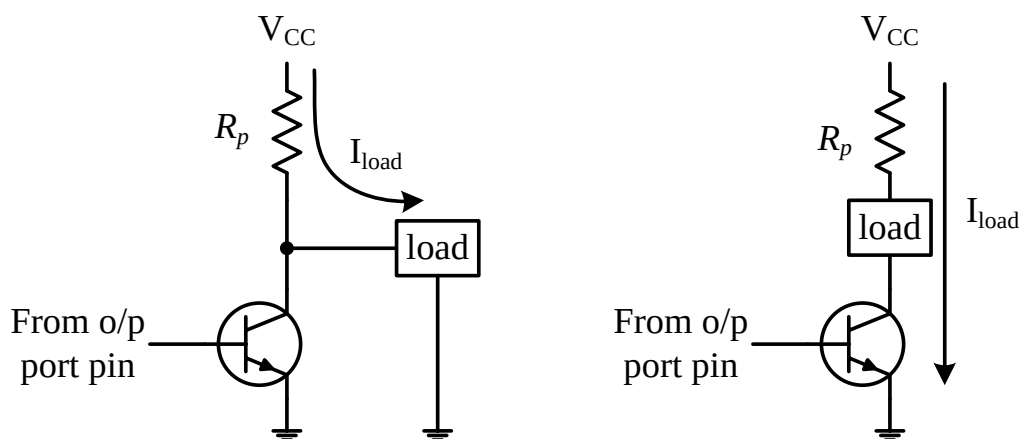


However, one must:

- (i) enable the pull-up resistor of Port D by setting PUPD; and
- (ii) examine the output voltage and fan-out current that is sufficient large to turn on the LED.
- (iii) if not, use buffer to solve the problem or choose a LED with electrical characteristics matches that of the port output.

The electrical characteristics of the ports are given in the table on next page.

It is important to note that the current driving capability of the I/O port is limited (see the Type Bn in the table). If the I/O port is used to drive a load that requires a higher voltage and/or a large current, for example a motor (a wheel of a toy car), a transistor circuit is required to source (supply) the current to the load.



D.C. ELECTRICAL CHARACTERISTICS**Table 19-3. Normal I/O PAD DC Electrical Characteristics**VDDP = 3.3 V $\pm$ 0.3 V, TA = 0 to 70 °C

Symbol	Parameters	Condition	Min	Typ	Max	Unit
VT+	Schmitt trigger, positive-going threshold	LVC MOS			2.0	V
VT-	Schmitt trigger, negative-going threshold	LVC MOS	0.8			
V _H	VT+ - VT-	Schmitt trigger	0.5	0.575	0.65	
I _{IH}	High level input current					μA
	Input buffer	V _{IN} = V _{DDP}	-10		10	
	Input buffer with pull-up		10	33	60	
I _{IL}	Low level input current					μA
	Input buffer	V _{IN} = V _{SS}	-10		10	
	Input buffer with pull-up		-60	-33	-10	
V _{OH}	High level output voltage ^(note)					V
	Type B6	I _{OH} = -6 mA	2.4			
	Type B8	I _{OH} = -8 mA				
	Type B10	I _{OH} = -10 mA				
	Type B12	I _{OH} = -12 mA				
V _{OL}	Low level output voltage ^(note)					V
	Type B4	I _{OH} = 4 mA			0.4	
	Type B6	I _{OH} = 6 mA				
	Type B8	I _{OH} = 8 mA				
	Type B10	I _{OH} = 10 mA				
	Type B12	I _{OH} = 12 mA				
I _{DS}	Stop current	V _{IN} = V _{SS} or V _{DD}		5		μA @25°C
I _{DD}	Operating current				1.2	mA/MHz

NOTE: Type B4 means 4mA output driver cell, and Type B8 means 8mA output driver cells.

Timing consideration:

The speed of data transfer via the I/O port is dependent upon the design of the port circuit itself.

For most design, the I/O port speed is not high (10 to 20 MHz) for the sake of reducing power consumption, particularly for hand-held battery operated devices.

If high speed data transfer is required, designs in 7.4.1 and 7.4.2 can be considered.

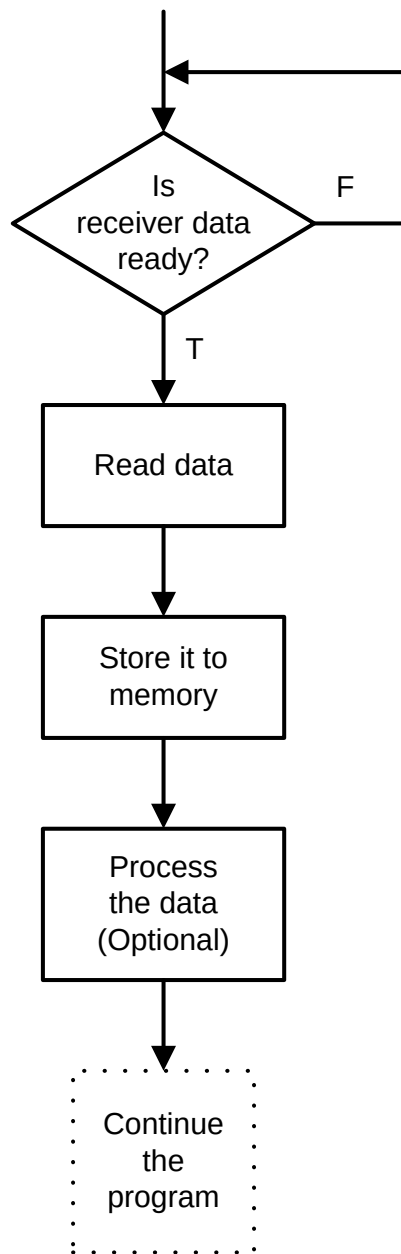
7.6 I/O programming

7.6.1 Status-polling I/O

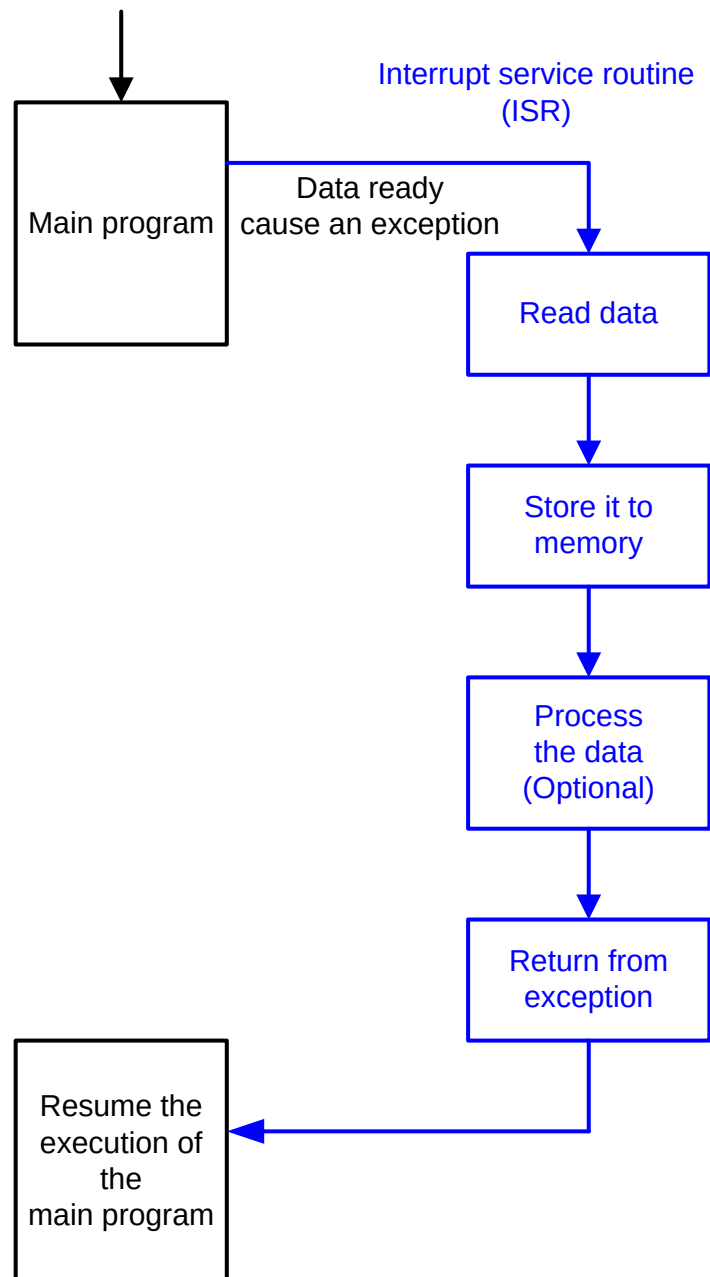
- When Handshaking is used in the I/O interface design, data transfer only takes place when certain conditions fulfill.
- Polling I/O is also known as programmed I/O.
- The program continues to interrogate readiness of peripheral interface for data transfer, the processor **cannot** do other things during this period.
- This usually involves of reading a status bit (for example data_ready bit) from the SR of an interface chip.
- This approach is inefficient since processing is wasted on checking some status bits.

7.6.2 Interrupt-driven I/O

- The interrupt-driven I/O allows the processor to perform other tasks when data is not ready for transfer.
- For an interrupt-driven I/O, the peripheral interface sends an interrupt request to the processor when a data byte is ready for transfer.
- Upon receiving the interrupt request, the processor responses (after checking the interrupt priority) by means of executing an interrupt service routine (ISR) to perform the data transfer.
- The processor resumes its normal operation until the next interrupt request is received.
- Processor does not need to waste time on check the status of the device and hence runs with higher efficiency.
- Figure below compares the logic flow of the polling scheme and the interrupt-driven scheme.



(a) Status polling



(b) Interrupt-driven I/O

Status polling versus interrupt-driven I/O

7.7 Programming in C-code

Assembly code:

```
.equ  PconG, 0x01d20040 /* Port G Config. Port */
.equ  PdatG, 0x01d20044 /* Port G Data Port */
.equ  PconD, 0x01d2001c /* Port D Config. Port */
.equ  PdatD, 0x01d20020 /* Port D Data Port */

.global _start
.text
_start:
    LDR    R0, =0x5555 /* Setup the config. value to Port D */
    LDR    R1, =0x0000 /* Setup the config. value to Port G */
    LDR    R2, =PconD   /* R2 point to Port D config. register */
    LDR    R3, =PconG   /* R3 point to Port G config. register */
    STRH   R0, [R2]     /* Set Port D as O/P: PConG is 16 bits */
    STRH   R1, [R3]     /* Set Port G as I/P: PConD is 16 bits */
    LDR    R2, =PdatD   /* R2 point to PdatD - 8 bits */
    LDR    R3, =PdatG   /* R2 point to PdatG - 8 bits */

loop:  LDR    R4, [R3]   /* Read the dil switch */
       STR    R4, [R2]   /* O/P the data to LED */
       B      loop

.end
```

In C-code:

```
#define rPCOND      (*(volatile unsigned *)0x1d2001c)
#define rPDATD      (*(volatile unsigned *)0x1d20020)
#define rPCONG      (*(volatile unsigned *)0x1d20040)
#define rPDATG      (*(volatile unsigned *)0x1d20044)
rPconG = 0x5555;    // Set Port G as I/P: PConG is 16 bits
rPconD = 0x0000;    // Set Port D as O/P: PConD is 16 bits
Data[1] = rPdatG;   // Read PortG data and store in Data[1]
rPdataD = Data[1];  // Send I/P data to O/P port D
```

Below is a program fragment, showing the port setting of the development board, extracted from 44blib.c

This is for illustrating how to set up the port for this board only. You may to change the setting for your own design!!!

```
//-----PORTS----- //
// These settings are only for initializing the on-board devices. //
// And requires modifications if external circuit is used. //

void Port_Init(void)
{
    //CAUTION:Follow the configuration order for setting the ports.
    // 1) setting value
    // 2) setting configuration register
    // 3) configure pull-up resistor.

    //16bit data bus configuration

    // PORT A GROUP
    // BIT    9    8    7    6    5    4    3    2    1    0
    //      A24 A23 A22 A21 A20 A19 A18 A17 A16 A0
    //      0    1    1    1    1    1    1    1    1    1
    rPCONA = 0x1ff;

    // PORT B GROUP
    // BIT    10    9    8    7    6    5    4    3    2    1    0
    //      /CS5 /CS4 /CS3 /CS2 /CS1 GPB5 GPB4 /SRAS /SCAS SCLK SCKE
    //      EXT  NIC  USB  IDE  SMC  NC   NC   Sdram Sdram Sdram Sdram
    //      1    1    1    1    1    0    0    1    1    1    1
    rPDATB = 0x7ff;           // P9-LED1 P10-LED2
    rPCONB = 0x1cf;

    // PORT C GROUP
    // BUSWIDTH=16
    // PC 15  14    13    12    11    10    9    8
    //      I   I    RXD1 TXD1 I     I     I     I
    //      NC  NC    Uart1 Uart1 NC   NC   NC   NC
    //      00  00    11    11    00   00   00   00

    // PC 7    6    5    4    3    2    1    0
    //      I    I    I    I    I    I    I    I
    //      NC   NC   NC   NC   IISCLK IISDI IISDO IISLRCK
    //      00   00   00   00   11     11   11   11
    rPDATC = 0xff00;
    rPCONC = 0x0ff0ffff;
    rPUPC = 0x30ff;           //PULL UP RESISTOR should be enabled to I/O

    // PORT D GROUP
    // PORT D GROUP(I/O OR LCD)
    // BIT 7    6    5    4    3    2    1    0
    //      VF    VM VLINE VCLK VD3 VD2 VD1 VD0
    //      00    00  00    00  00  00  00  00
}
```

```

rPDATD= 0xff;
rPCOND= 0xaaaa;
rPUPD = 0x0;
// These pins must be set only after CPU's internal LCD controller is enable

// PORT E GROUP
// Bit   8       7       6       5       4       3       2       1       0
// CODECLK LED4 LED5 LED6 LED7 BEEP RXD0 TXD0 LcdDisp
//      10      01      01      01      01      01      10      10      01
rPDATE  = 0x1ff;
rPCONE  = 0x25529;
rPUPE = 0x6;

// PORT F GROUP
// Bit   8       7       6       5       4       3       2       1       0
// IISCLK IISDI IISDO IISLRCK Input Input Input IICSDA IICSL
//      100     100     100     100     00      00      00      10      10
rPDATF = 0x0;
rPCONF = 0x252a;
rPUPF = 0x0;

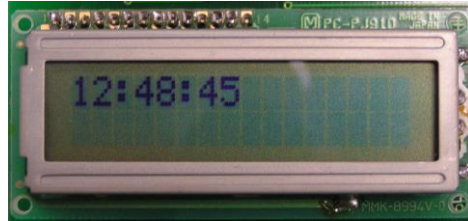
// PORT G GROUP
// BIT 7       6       5       4       3       2       1       0
// INT7 INT6 INT5 INT4 INT3 INT2 INT1 INT0
// S3 S4 S5 S6 NIC EXT IDE USB
// 11 11 11 11 11 11 11 11
rPDATG = 0xff;
rPCONG = 0xffff;
rPUPG = 0x0; //should be enabled
rSPUCR = 0x7; //D15-D0 pull-up disable

/* Non Cache area */
rNCACHBE0=((Non_Cache_End>>12)<<16)|(Non_Cache_Start>>12);
/* Low level default */
rEXTINT=0x0;
}

```

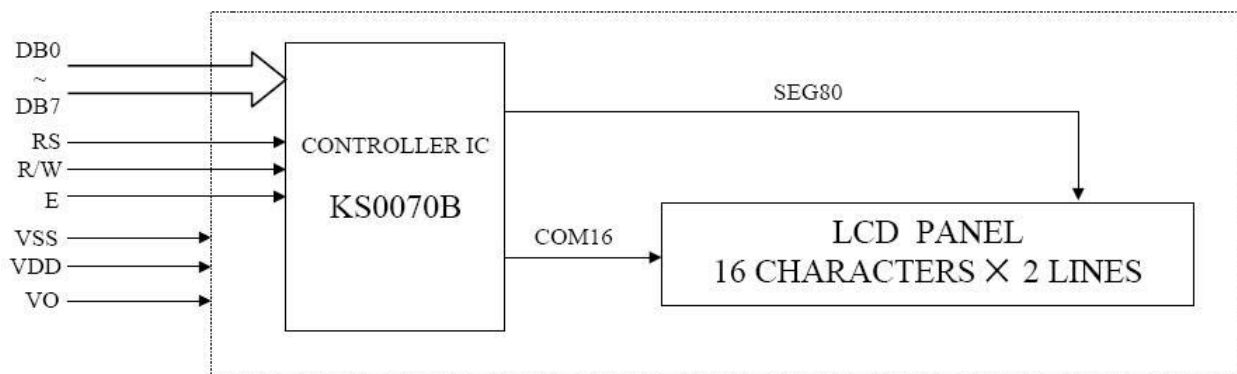
7.8 Design example

Interfacing a two-row character LCD to the ARM board for displaying the RTC:



MTC-C162DPRN-2N

Block diagram



Pin	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LCD Pin	Vss	Vdd	Vo	RS	R/W	E	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7
Function														
Port Name	0V	5V	0V	PG0	PG1	PG2	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7

Pins connections between LCD and the Ports D & G

Control and Display Command

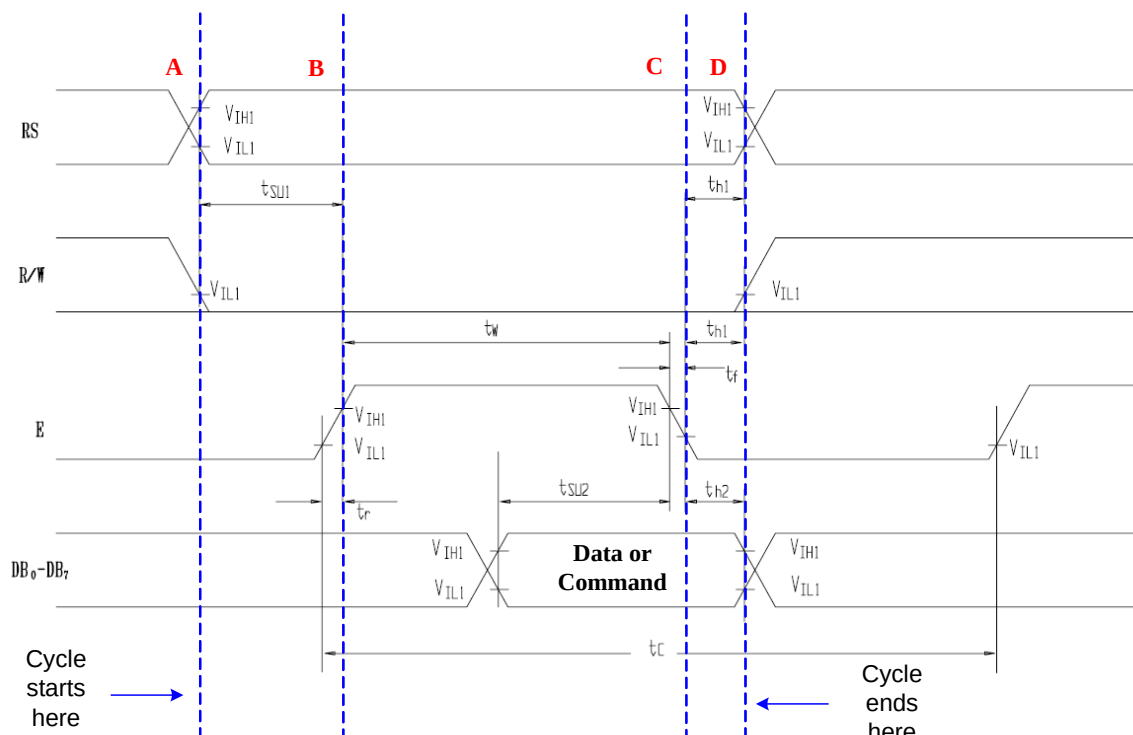
Command	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	Execution Time (f _{osc} = 250kHz)	Remark															
DISPLAY CLEAR	L	L	L	L	L	L	L	L	L	H	1.64ms																
RETURN HOME	L	L	L	L	L	L	L	L	H	X	1.64ms	Cursor move to first digit															
ENTRY MODE SET	L	L	L	L	L	L	L	H	I/D	SH	42μs	- I/D : Set cursor move direction<table><tr><td rowspan="2">I/D</td><td>H</td><td>Increase</td></tr><tr><td>L</td><td>Decrease</td></tr></table> - SH : Specifies shift of display<table><tr><td rowspan="2">SH</td><td>H</td><td>Display is shifted</td></tr><tr><td>L</td><td>Display is not shifted</td></tr></table>	I/D	H	Increase	L	Decrease	SH	H	Display is shifted	L	Display is not shifted					
I/D	H	Increase																									
	L	Decrease																									
SH	H	Display is shifted																									
	L	Display is not shifted																									
DISPLAY ON/OFF	L	L	L	L	L	L	H	D	C	B	42μs	- Display<table><tr><td rowspan="2">D</td><td>H</td><td>Display on</td></tr><tr><td>L</td><td>Display off</td></tr></table> - Cursor<table><tr><td rowspan="2">C</td><td>H</td><td>Cursor on</td></tr><tr><td>L</td><td>Cursor off</td></tr></table> - Blinking<table><tr><td rowspan="2">B</td><td>H</td><td>Blinking on</td></tr><tr><td>L</td><td>Blinking off</td></tr></table>	D	H	Display on	L	Display off	C	H	Cursor on	L	Cursor off	B	H	Blinking on	L	Blinking off
D	H	Display on																									
	L	Display off																									
C	H	Cursor on																									
	L	Cursor off																									
B	H	Blinking on																									
	L	Blinking off																									
SHIFT	L	L	L	L	L	H	S/C	R/L	X	X	42μs	<table><tr><td rowspan="2">S/C</td><td>H</td><td>Display shift</td></tr><tr><td>L</td><td>Cursor move</td></tr></table> <table><tr><td rowspan="2">R/L</td><td>H</td><td>Right shift</td></tr><tr><td>L</td><td>Left shift</td></tr></table>	S/C	H	Display shift	L	Cursor move	R/L	H	Right shift	L	Left shift					
S/C	H	Display shift																									
	L	Cursor move																									
R/L	H	Right shift																									
	L	Left shift																									
SET FUNCTION	L	L	L	L	H	DL	N	F	X	X	42μs	<table><tr><td rowspan="2">DL</td><td>H</td><td>8 bits interface</td></tr><tr><td>L</td><td>4 bits interface</td></tr></table> <table><tr><td rowspan="2">N</td><td>H</td><td>2 line display</td></tr><tr><td>L</td><td>1 line display</td></tr></table> <table><tr><td rowspan="2">F</td><td>H</td><td>5 X 10 dots</td></tr><tr><td>L</td><td>5 X 7 dots</td></tr></table>	DL	H	8 bits interface	L	4 bits interface	N	H	2 line display	L	1 line display	F	H	5 X 10 dots	L	5 X 7 dots
DL	H	8 bits interface																									
	L	4 bits interface																									
N	H	2 line display																									
	L	1 line display																									
F	H	5 X 10 dots																									
	L	5 X 7 dots																									
SET CG RAM ADDRESS	L	L	L	H	CG RAM address (corresponds to cursor address)						42μs	CG RAM Data is sent and received after this setting															
SET DD RAM ADDRESS	L	L	H	DD RAM address						42μs	DD RAM Data is sent and received after this setting																
READ BUSY FLAG & ADDRESS	L	H	BF	Address Counter used for both DD & CG RAM address						0μs	<table><tr><td rowspan="2">BF</td><td>H</td><td>Busy</td></tr><tr><td>L</td><td>Ready</td></tr></table> - Reads BF indication internal operating is being performed - Reads address counter contents	BF	H	Busy	L	Ready											
BF	H	Busy																									
	L	Ready																									
WRITE DATA	H	L	Write Data						46μs	Write data into DD or CG RAM																	
READ DATA	H	H	Read Data						46μs	Read data from DD or CG RAM																	

X : Don't care

In order to operate the LCD module correctly, a series of command should be sent to the LCD for proper initialization

Sending the control signal, command and data to the LCD must fulfill the timing requirement as shown below. **Each cycle can only send one command or data.** The control signals change at 4 time points (A, B, C and D) and the delay time gap between these time points can be realized by executing a simple delay function, for example the `DelayMs( )` function. (*Note the signal level of RS signifies $DB_0 \sim DB_7$ to be either command or data.*)

Characteristic	Symbol	Min.	Typ.	Max.	Unit	Test pin
E cycle time	t_C	500	---	---	ns	E
E rise time	t_r	---	---	25	ns	E
E fall time	t_f	---	---	25	ns	E
E pulse width (High, Low)	t_W	220	---	---	ns	E
R/W and RS set-up time	t_{SU1}	40	---	---	ns	R/W, RS
R/W and RS hold time	t_{h1}	10	---	---	ns	R/W, RS
Data set-up time	t_{SU2}	60	---	---	ns	$DB_0 \sim DB_7$
Data hold time	t_{h2}	10	---	---	ns	$DB_0 \sim DB_7$



By writing a “1” to the port pin, it will cause the pin’s output to be 1 or otherwise “0”.

Setting a port with “1”, “0” and “1” will generate a negative pulse and the pulse width can be controlled by inserting a delay loop.

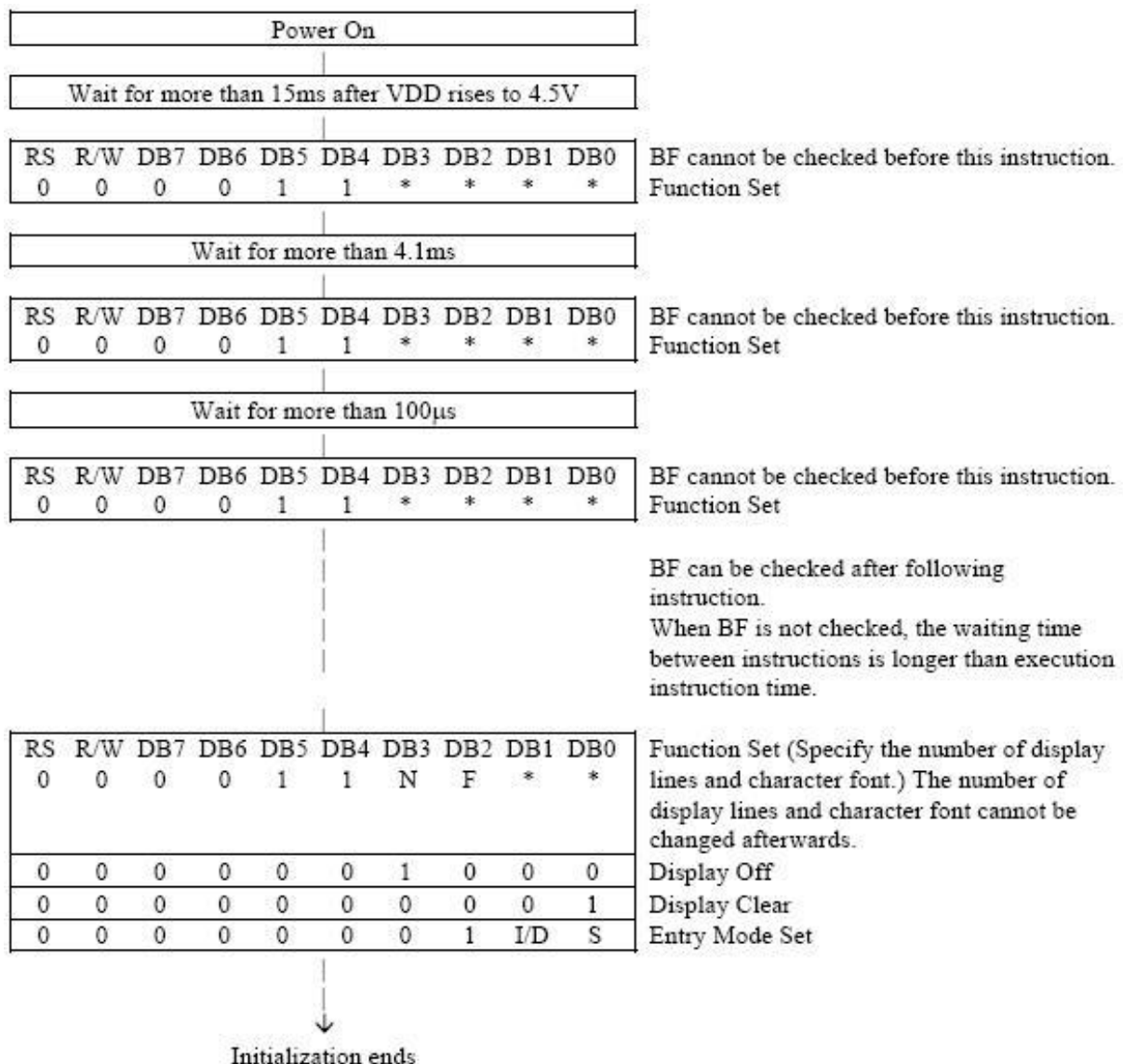
LCD - Initializing by Internal Reset Circuit:

Before displaying characters on the LCD, it must be initialized first.

The KS0070B automatically initializes (resets) when the power is on using the internal reset circuit. The following instructions are executed in initialization. The busy flag is kept in busy state (BF=1) until initialization ends. The busy state is 10ms after VDD rises to 4.5V.

- (1) Display Clear
- (2) Function Set
 - DL = 1 : 8-bit interface data
 - N = 0 : 1-line display
 - F = 0 : 5x7-dot character font
- (3) Display On/Off Control
 - D = 0 : Display Off
 - C = 0 : Cursor Off
 - B = 0 : Blink Off
- (4) Entry Mode Set
 - I/D = 1 : +1 (Increment)
 - S = 0 : No Shift

Initializing by instruction for more display options:



Flow of program:

- Write a function to send Command to LCD.
- Write a function to send Data to LCD.
- Both functions are almost the same except for command and data.
- The function need to simulate the correct timing for E, RS and R/W lines by inserting delaysms() function
- Set port function
- Initialize the LCD by sending a sequence of commands
- Send ASCII data for displaying on the LCD.