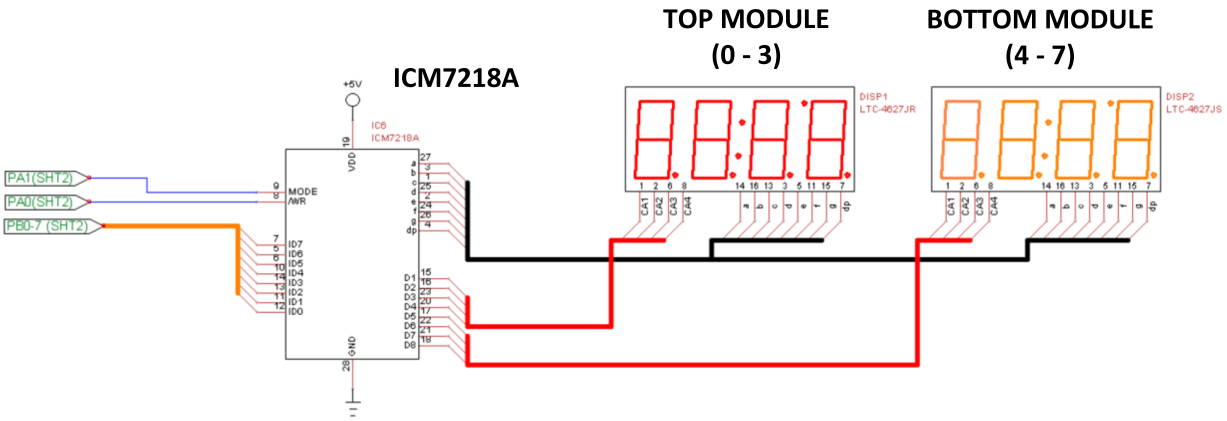
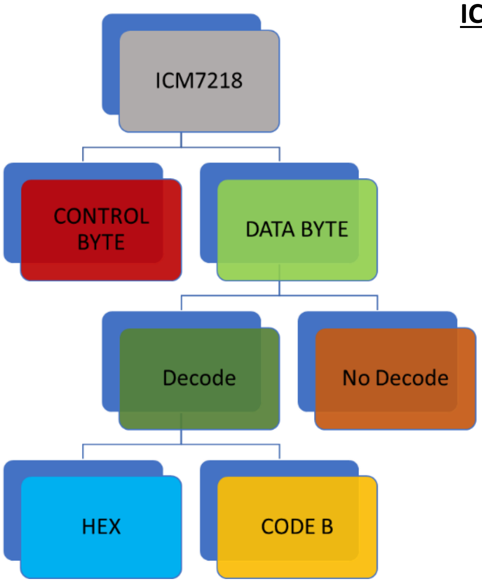
7 Segment Displays and ICM7218 Controller



ICM7218 Controller



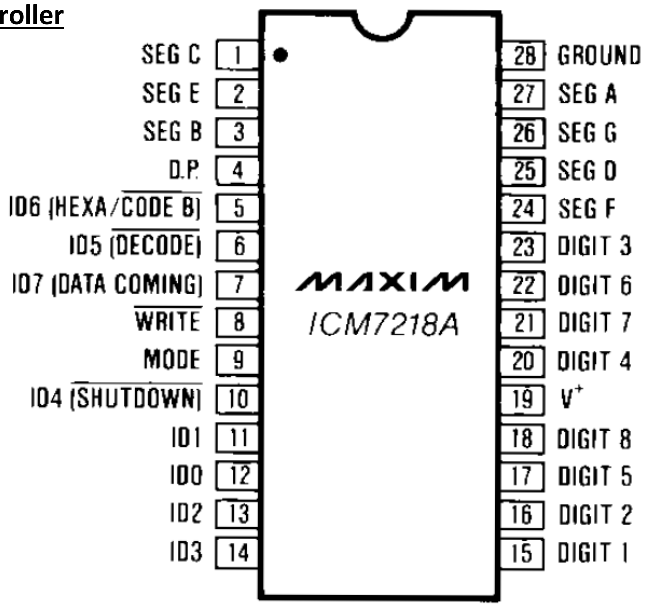
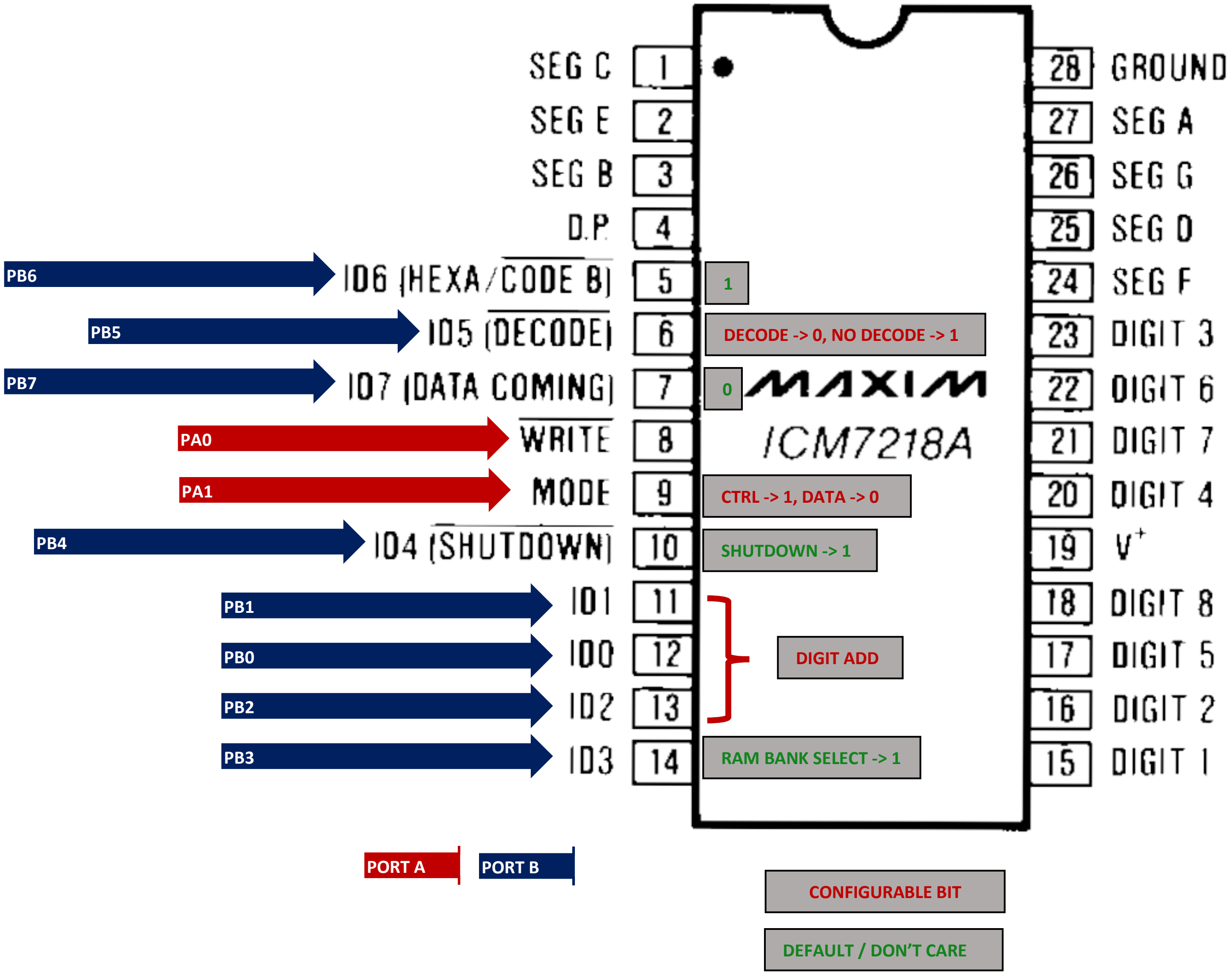


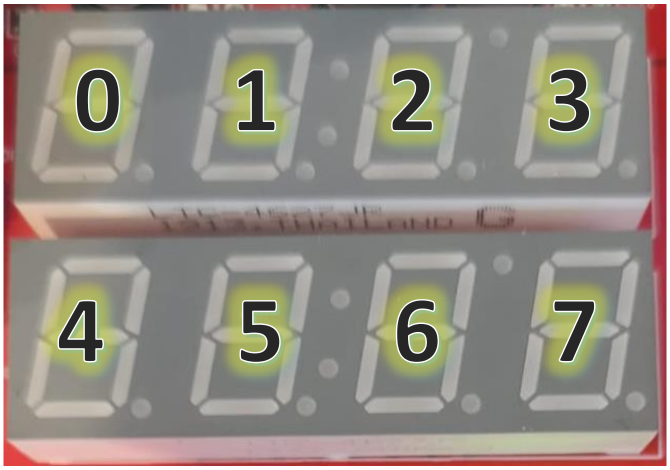
Table 1. Input Definitions, ICM7218A and ICM7218B
Note: Pin Configurations for the ICM7218A/B are shown on last page.

INPUT	PIN	STATE	FUNCTION
WRITE	8	High Low	Input Not Loaded Into Memory Input Loaded Into Memory
MODE	9	High Low	Loads Control Word on WR Loads Input Data on WR
ID0-ID2, DIGIT ADDRESS	12, 11, 13	High Low	Loads "one" Loads "zero"
ID3, BANK SELECT	14	High Low	Select RAM Bank A (Hex or Code B Select RAM Bank B Data only)
ID4, SHUTDOWN (MODE High)	10	High Low	Normal Operation Shutdown
ID5, DECODE/NO DECODE (MODE High)	6	High Low	No Decode Decode
ID6, HEX/CODE B (MODE High)	5	High Low	Hexadecimal Decoding Code B Decoding
ID7, DATA COMING (MODE High)	7	High Low	Data Coming (control word) No Data Coming (control word)
ID0-ID7, INPUT DATA (MODE Low)	5-7, 10-14	High Low	Loads "one" (Note 1) Loads "zero" (Note 1)

SUMMARY



ADDRESS



Writing Data to the Controller

The interaction with the controller uses two ports from our micro:

- Port B: To present the control byte or data byte (**PB0 – PB7**)

- PortA: To specify if the operation is a control byte or data byte int the MODE pin (**PA1**) and to *latch* the data present in Port B by driving the WRITE pin LOW (**PA0**) for a brief moment and then pulling it back HIGH

There are two possible ways of writing data do the led display controller:

- Write a byte into a specific address: Involves sending a control byte followed by a data byte. Both the control byte and data byte need to be *latched* accordingly.

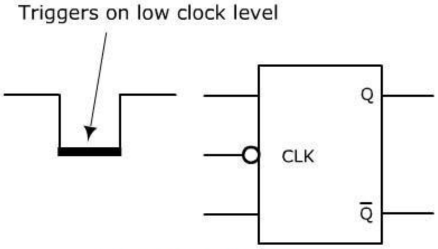
- Write 8 consecutive bytes through address 0 to 7: Involves sending a control byte followed by eight data bytes. The control byte and all the data bytes need to be *latched* accordingly.

We will mainly focus on the first option, *writing a single byte* at a time.

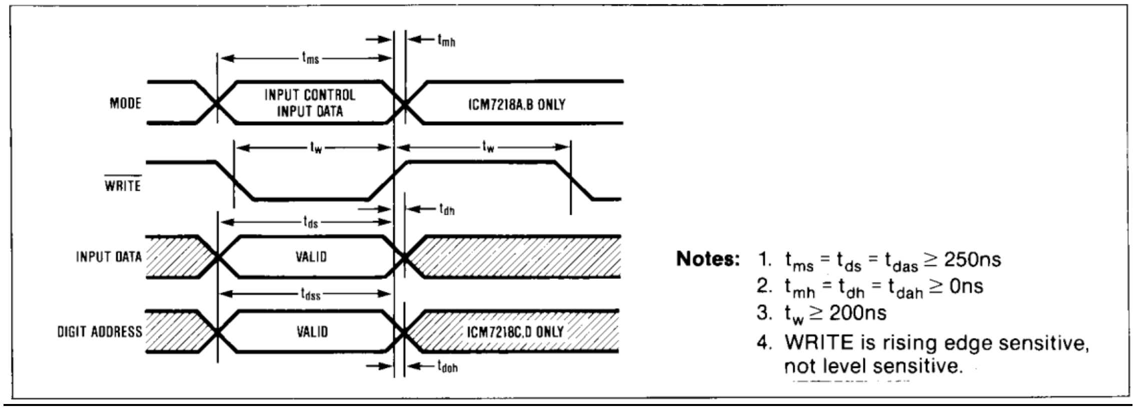
Latch Data or Control Byte

Send a low level pulse on WRITE pin

- ○ RESET WRITE PIN (PA0)
- ○ SET WRITE PIN (PA0)


Triggers on low clock level

Low Level Triggering



Notes:
1. $t_{ms} = t_{ds} = t_{das} \geq 250ns$
2. $t_{mh} = t_{dh} = t_{dah} \geq 0ns$
3. $t_w \geq 200ns$
4. WRITE is rising edge sensitive, not level sensitive.

This diagram shows that in order to latch the data properly, the WRITE pin has to be LOW for at least 200[ns] before it gets pull HIGH again. Considering our default clock speed of 8MHZ, each cycle is 125[ns] so with 2 or 3 cycles, we meet the requierements. The BCLR assembly instruction takes 4 cycles so we are good because 4 X 125[ns] = 500[ns]. However, if we set our clock to 24MHZ, each cycle is about 42[ns], therefore 4 x 42[ns] = 167[ns], we may start to get into the verge or latching too fast. This gets worse if we increase the clock speed even higher.

Process To Write Single Byte

We will describe the step by step process of writing a single byte into the controller. As we have mentioned, this involves two steps: sending a control byte and sending a data byte:

A. Send a Control Byte

- Specify Control Mode
- Set addres: 0 – 7
- Specify other settings
- SET Mode BIT (PA1)
- Latch Control Byte on the WRITE pin (PA0)

Port B configuration bits:

1. Enter the Data address into **BIT0 - BIT2**. A good idea is to mask the address to a maximum of **0x7** (**111b**) to avoid problems due to a wrong address passed in.
2. Select RAM **Bank A** (You could try Bank B to see what happens) by **SETTING** the **BIT3**.
3. We want normal operation (No Shutdown), so we **SET BIT4**.
4. We select **Decode (0)** for writing in **HEX** and **No Decode(1)** for writing ***custom characters*** (individual segments). This gets configured by **SETTING** or **RESETTING BIT5**
5. We Select Decode as **HEX** (Your could try decode as CODEB to see what happens) by **SETTING BIT6.** Please note that this setting is irrelevant if you selected *No Decode* in step 4.
6. As we said, the simplest way to write to the controller is to write 1 byte at the time (1 control byte + 1 data byte). In order to do this, we select ***No Data Coming*** by **RESETTING BIT7**. If we wanted to write a stream of the ***7 bytes all at once***, we would **SET BIT7**

SINGLE/M	HEX/CODE B	DEC/NO DEC	NORMAL 1	BANK A/B	ADDR2	ADDR1	ADDR0
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0**

BIT7: 0 -> No Data coming, 1-> 8 Bytes coming

BIT6: 0 -> Decode as Code B, 1-> Decode as HEX

BIT5: 0 -> Decode, 1-> No decode (custom)

BIT4: 0 -> Shutdown, 1 -> Normal

BIT3: 0 -> Bank B, 1 -> Bank A,

BIT2 – BIT0: Address

Port A configuration bits:

1. **SET BIT1** to select Control Mode on Mode PIN.
2. **RESET BIT0** to pull the **WRITE** pin **LOW** and insert a delay if necessary (a couple of microseconds will do it)
3. **SET BIT0** to finish the pulse and pull the pin **HIGH**, now the control byte is latched

B. Send a Decoded Data Byte (Hex or Code B)

- Present Data on Port B, **BIT0 – BIT3**
- Specify Decode
- **SET(1) BIT7** if **NO Decimal Point**, **RESET(0) BIT7** if Decimal Point is to be **printed**
- **RESET** Mode bit (PA1)
- Latch Data Byte on the WRITE pin (PA0)

Decode Mode: HEX or CODE B

ID3	ID2	ID1	ID0	HEXADECIMAL	CODE B
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	A	-
1	0	1	1	B	E
1	1	0	0	C	H
1	1	0	1	D	L
1	1	1	0	E	P
1	1	1	1	F	(Blank)

Port B configuration bits:

1. Clear Port B by writing a 0
2. Write the data into **BIT0 – BIT3**. Remember the data is in the range of 0000b – 1111b.
3. To write the decimal point segment leave **BIT7 RESET(0)**, to remove the decimal point, **SET BIT7(1).**
4. BIT4 – BIT6 are don't care when writing a decoded data byte.

DEC POINT	X	X	X	DATA3	DATA2	DATA1	DATA0
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0**

BIT7: 0 -> Decimal point, 1-> No decimal point

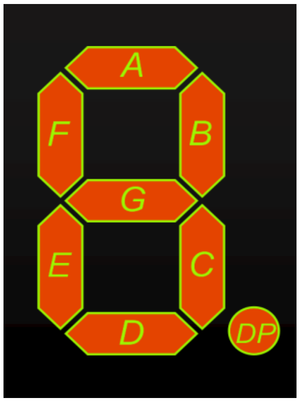
BIT3 – BIT0: Data

Port A configuration bits:

1. **RESET BIT1** to select Data Mode on Mode PIN.
2. **RESET BIT0** to pull the **WRITE** pin **LOW** and insert a delay if necessary (a couple of microseconds will do it).
3. **SET BIT0** to finish the pulse and pull the pin **HIGH**, now the control byte is latched

C. Send a No Decoded Data Byte (Custom segments)

- Present Data on Port B, **BIT0 – BIT6**
- Specify No Decode in the previous control byte
- **SET(1) BIT7** if **NO Decimal Point**, **RESET(0) BIT7** if Decimal Point is to be **printed**
- **RESET** Mode bit (PA1)
- Latch Data Byte on the WRITE pin (PA0)



No Decode Mode

ON/OFF	0/1	1/0	1/0	1/0	1/0	1/0	1/0	1/0
Data Input	**ID7**	**ID6**	**ID5**	**ID4**	**ID3**	**ID2**	**ID1**	**ID0**
Controller Segment	**Decimal Point**	**A**	**B**	**C**	**E**	**G**	**F**	**D**

All On: 0b01111111

All Off: 0b10000000

Port B configuration bits:

DP	SEG A	SEG B	SEG C	SEG E	SEG G	SEG F	SEG D
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0**

BIT7: 0 -> Decimal point, 1-> No decimal point

BIT3 – BIT0: Data

Port A configuration bits:

1. **RESET BIT1** to select Data Mode on Mode PIN.
2. **RESET BIT0** to pull the **WRITE** pin **LOW** and insert a delay if necessary (a couple of microseconds will do it).
3. **SET BIT0** to finish the pulse and pull the pin **HIGH**, now the control byte is latched