E-paper Display COG
Driver Interface Timing
for 1.44”,1.9”,2”,2.6” and 2.7” EPD with G2 COG and Aurora Mb Film

Description	**Detailed information to design a timing controller for 1.44”, 1.9”, 2”, 2.6” and 2.7” EPD with G2 COG and Aurora Mb film**
Date	**2015/07/27**
Doc. No.	**4P018-00**
Revision	**03**

	Design Engineering		
	Approval	**Check**	**Design**
	黃 2015.07.27 朝玄	黃 2015.07.27 朝玄	丁 2015.07.27 昭文

4F, No. 28, Chuangye Rd., Tainan Science Park, Tainan City 74144, Taiwan (R.O.C.)

Tel: +886-6-279-5399 Fax: +886-6-505-5300

Copyright

龍亭新技股份有限公司 Pervasive Displays Inc.

4F, No. 28, Chuangye Rd., Tainan Science Park, Tainan City 74144, Taiwan (R.O.C.)

Tel: +886-6-279-5399

http://www.pervasivedisplays.com

Table of Contents

I. Revision History

Version	Date	Page (New)	Section	Description
Ver. 01	2014/11/20	All	All	First issued
Ver. 02	2015/02/16	8	1.2	Modify the description
		11	1.3.3	Modify the pin description
		14	1.3.6	Modify the description of Note number
		15	1.3.6	Add description about hardware suggestion
		30 31 32 33 34	5.2	Add the description of “Turn on OE”
		37	6	Modify the description BORDER -> /BORDER_CONTROL
		25	4	Modify OE setting (0x02,0x40) -> (0x02,0x04)
		37	6	Remove OE off flow (0x02,0x05)
Ver. 03	2015/07/27	All	All	Modify PDi company address
		All	All	Revise some descriptions in document
		7	1.2	Modify Pin.8 assignment
		17-19	1.6	Register Index -> Command Index Register Data -> Command Data Modify the delay time from 10us to 80ns between /CS high and low
		23	4	Modify delay time of power on sequence Modify OE setting (0x02,0x04) -> (0x02,0x06)
		27-31	5.3	Remove checking BUSY when sending each data byte
		32	5.4	Add notes
		35	7	Section 1.3.6 move to Appendix
		34	6	Modify delay time of VGL voltage

II. Glossary of Acronyms

EPD	Electrophoretic Display (e-Paper Display)
EPD Panel	EPD
Tcon	Timing Controller
MCU	Microcontroller Unit
FPL	Front Plane Laminate (e-Paper Film)
SPI	Serial Peripheral Interface
COG	Chip On Glass, Driver IC. There are two models: Gen1 (G1) is EOL and Gen2 (G2) which is used in this document
PDI, PDi	Pervasive Displays Incorporated

1. General Description

1.1 Overview

This document explains the interface to the G2 COG Driver to operate the EPD for a Timing Controller based solution using two pages of memory buffer. G2 COG Driver is the most recent EPD driving technology from PDi that offers new features such as breakage detection, lower inrush current, and a lower operation voltage.

The procedure to update display is

1. Store new pattern in memory buffer
2. Power on G2 COG Driver
3. Initialize G2 COG Driver
4. Update display stage by stage
5. Power off G2 COG Driver

Refer to the EPD controller in section 1.5 for the complete update cycle from Power On, Initialize, Update and Power off. To operate the EPDs for the best sharpness and performance, each update of the panel is divided into a series of stages before the display of the new image pattern is completed. During each stage, frame updates with intermediate image patterns are repeated for a specified period of time. The number of repeated frame updates during each stage is dependent on the Timing Controller speed. After the final stage, the new pattern is displayed.

Section 1 is an overview and contains supporting information such as the overall theory for updating an EPD, SPI timing for PDI's EPDs, as well as current profiles.

Section 2 describes a method to write to memory buffer. Previously updated and new patterns are stored in the memory buffer to compare the old and new image patterns during the update.

Section 3 describes how to power on the G2 COG Driver which consists of applying a voltage and generating the required signals for /CS and /RESET.

Section 4 describes the steps to initialize the G2 COG Driver.

Section 5 describes the details on how to update the EPD from the memory buffer, create a line of data, update in stages.

Section 6 describes how to power off the G2 COG Driver, and discharge voltage from EPD to ground, make sure there is no voltage remains in EPD.

1.2 Input Terminal Pin Assignment

No	Signal	I/O	Connected to	Function
1	/CS	I	TCon	Chip Select. Low enable
2	BUSY	O	TCon	When BUSY = HIGH, EPD stays in busy state that EPD ignores any input data from SPI.
3	ID	I	Ground	Set SPI interface
4	SCLK	I	TCon	Clock for SPI
5	SI	I	TCon	Serial input from host Timing Controller to EPD
6	SO	O	TCon	Serial output from EPD to host Timing Controller
7	/RESET	I	TCon	Reset signal. Low enable
8	BORDER_DRIVER or PWRON	-	BORDER or NC	For 1.44” & 2”, connect to BORDER. For 1.9”, 2.6” & 2.7”, keep open.
9	V_{CL}	C	Capacitor	-
10	C42P	C	NC	Not connected. These two pins are used only with G1 COG Drive IC.
11	C42M	C		
12	C41P	C	Charge-Pump Capacitor	-
13	C41M	C		-
14	C31M	C	Charge-Pump Capacitor	-
15	C31P	C		-
16	C21M	C	Charge-Pump Capacitor	-
17	C21P	C		-
18	C16M	C	Charge-Pump Capacitor	-
19	C16P	C		-
20	C15M	C	Charge-Pump Capacitor	-
21	C15P	C		-
22	C14M	C	Charge-Pump Capacitor	-
23	C14P	C		-

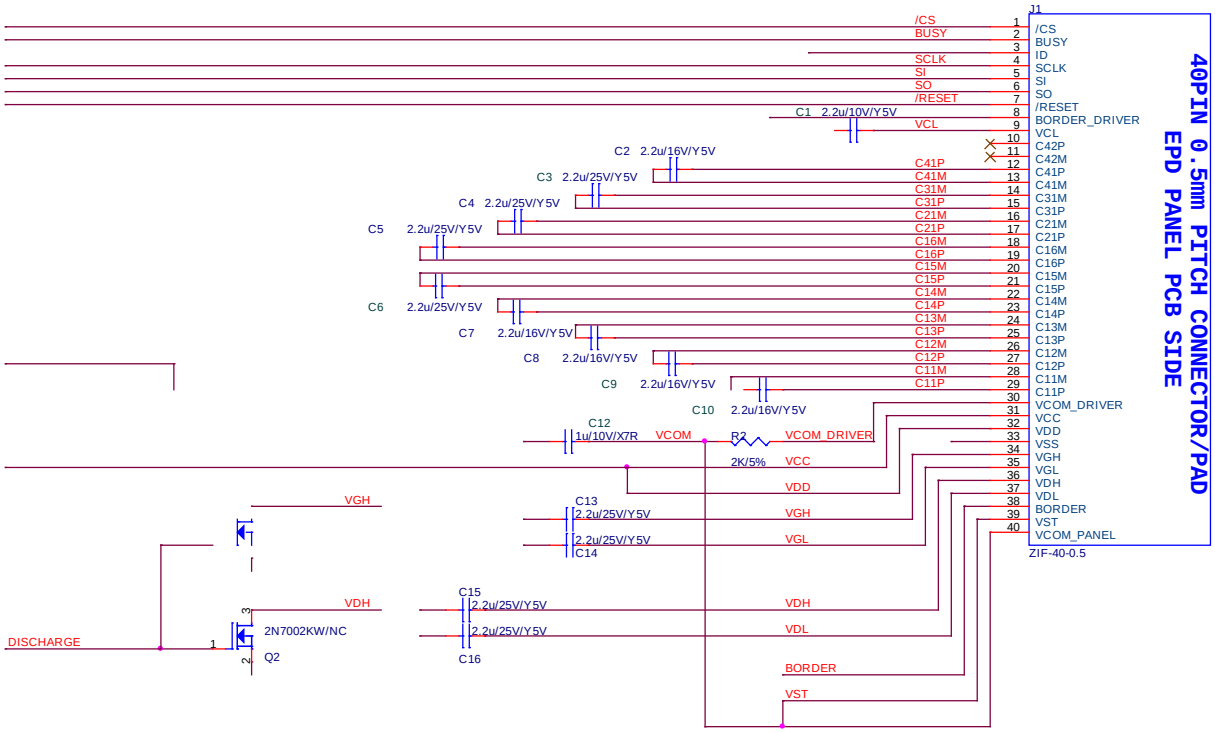
No	Signal	I/O	Connected to	Function
24	C13M	C	Charge-Pump Capacitor	-
25	C13P	C		-
26	C12M	C	Charge-Pump Capacitor	-
27	C12P	C		-
28	C11M	C	Charge-Pump Capacitor	-
29	C11P	C		-
30	V_{COM_DRIVER}	RC	Resistor & Capacitor	The duty cycle of V_{COM_DRIVER} can adjust V_{COM} voltage from source driver IC
31	V_{CC}	P	V_{CC}	Power supply for analog part of source driver
32	V_{DD}	P	V_{DD}	Power supply for digital part of source driver
33	V_{SS}	P	Ground	-
34	V_{GH}	C	Capacitor	-
35	V_{GL}	C	Capacitor	-
36	V_{DH}	C	Capacitor	-
37	V_{DL}	C	Capacitor	-
38	BORDER	I	-	For 1.44” & 2”, connect to BORDER_DRIVER For 2.7”, connect to V_{DL} via control circuit for white frame border For 1.9” & 2.6”, not connected
39	V_{ST}	P	V_{COM_PANEL}	-
40	V_{COM_PANEL}	C	Capacitor	V_{COM} to panel

Note:

1. **I**: Input, **O**: Output, **C**: Capacitor, **RC**: Resistor and Capacitor, **P**: Power, **NC**: Not connected (Do not SMT)
2. Recommend to use an independent SPI bus to control the EPD.
3. Around the active area of the EPD is a 0.5mm width blank area called the BORDER. It’s connected to VDL (-13V ~ -14V) to keep the border white. After EPD updates with the constant voltage, the border color may degrade to a gray level that is not as white as the active area. Reset the Border per screen update to avoid this phenomenon.

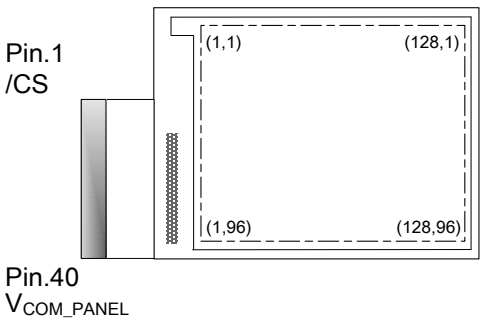
1.3 Reference Circuit

1.3.1 1.44 inch EPD Reference Circuit



Note :

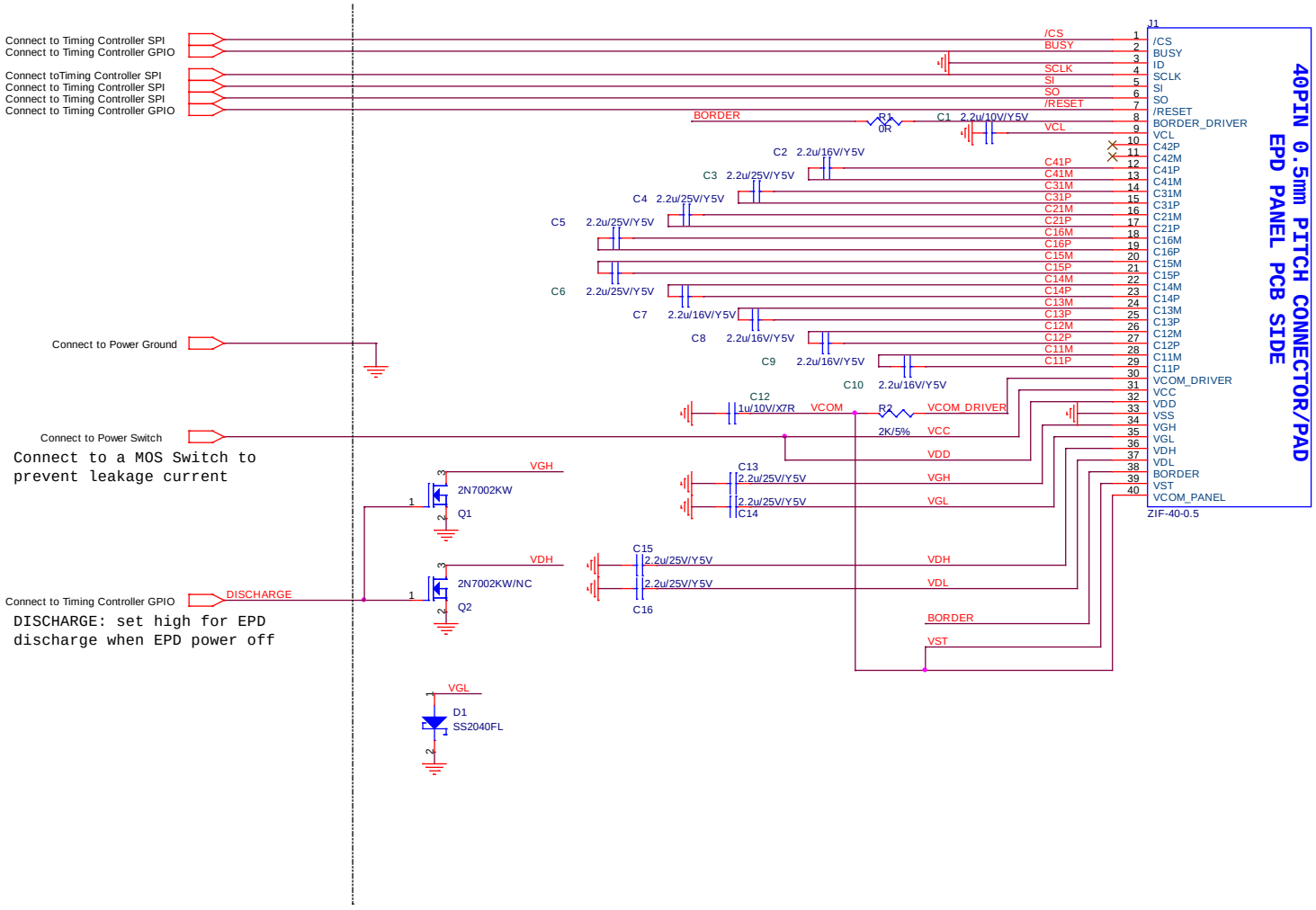
1. V_{DD} and V_{CC} must be discharged promptly after power off
2. 1.44" Pin.1 location



3. R1 is connected.

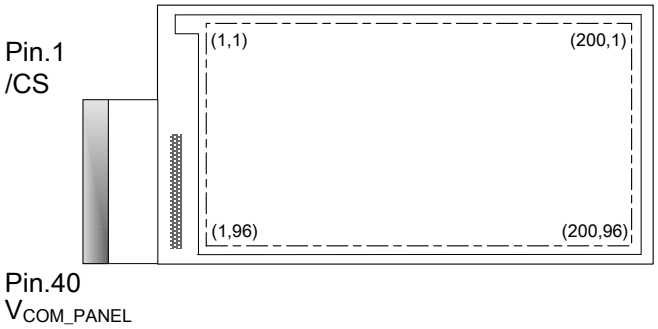
1.3.2 2 inch EPD Reference Circuit

INPUT



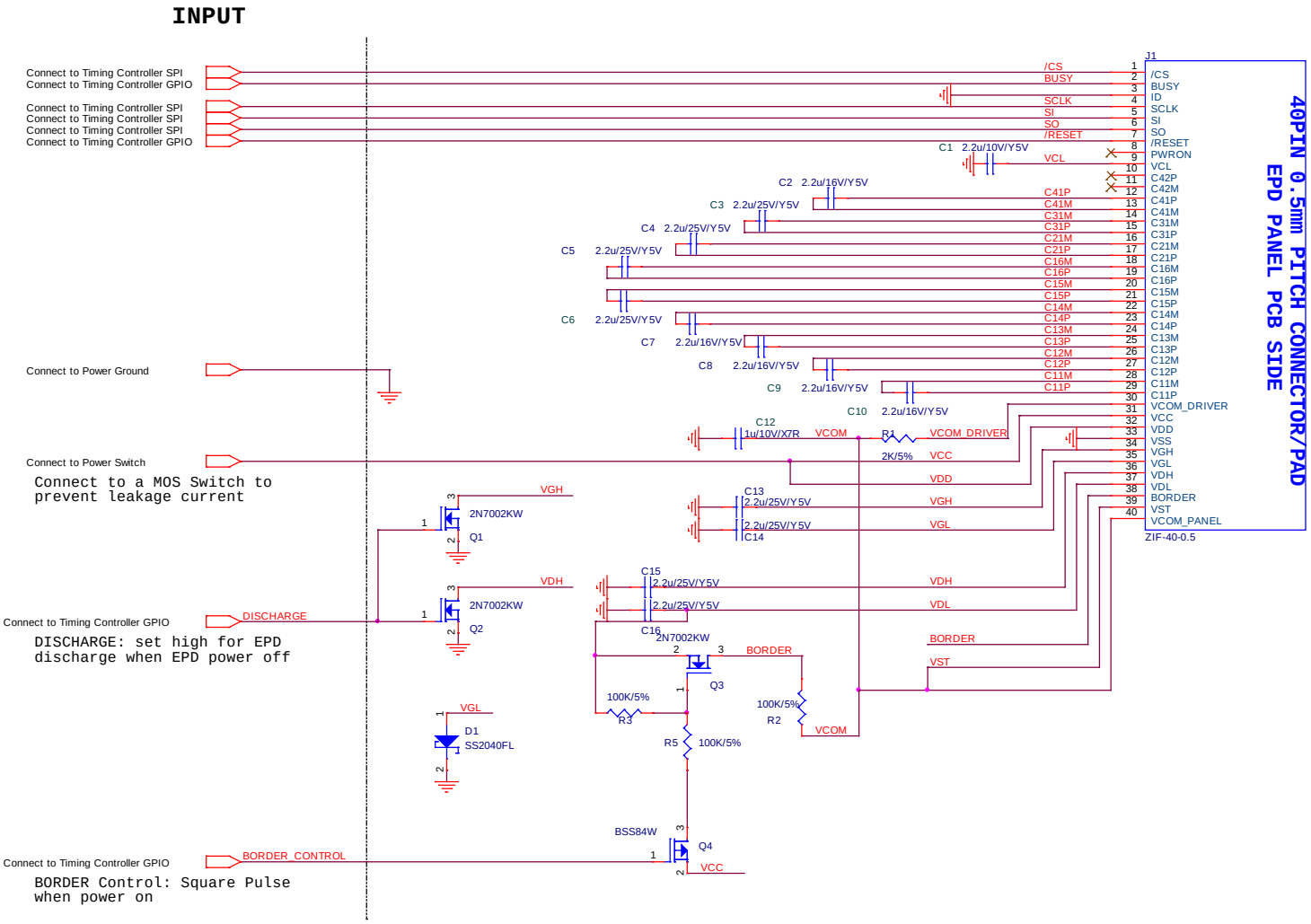
Note :

1. V_{DD} and V_{CC} must be discharged promptly after power off
2. 2" Pin.1 location



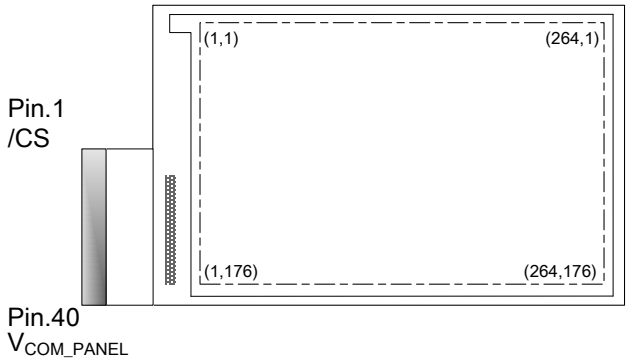
3. R1 is connected.

1.3.3 2.7 inch EPD Reference Circuit

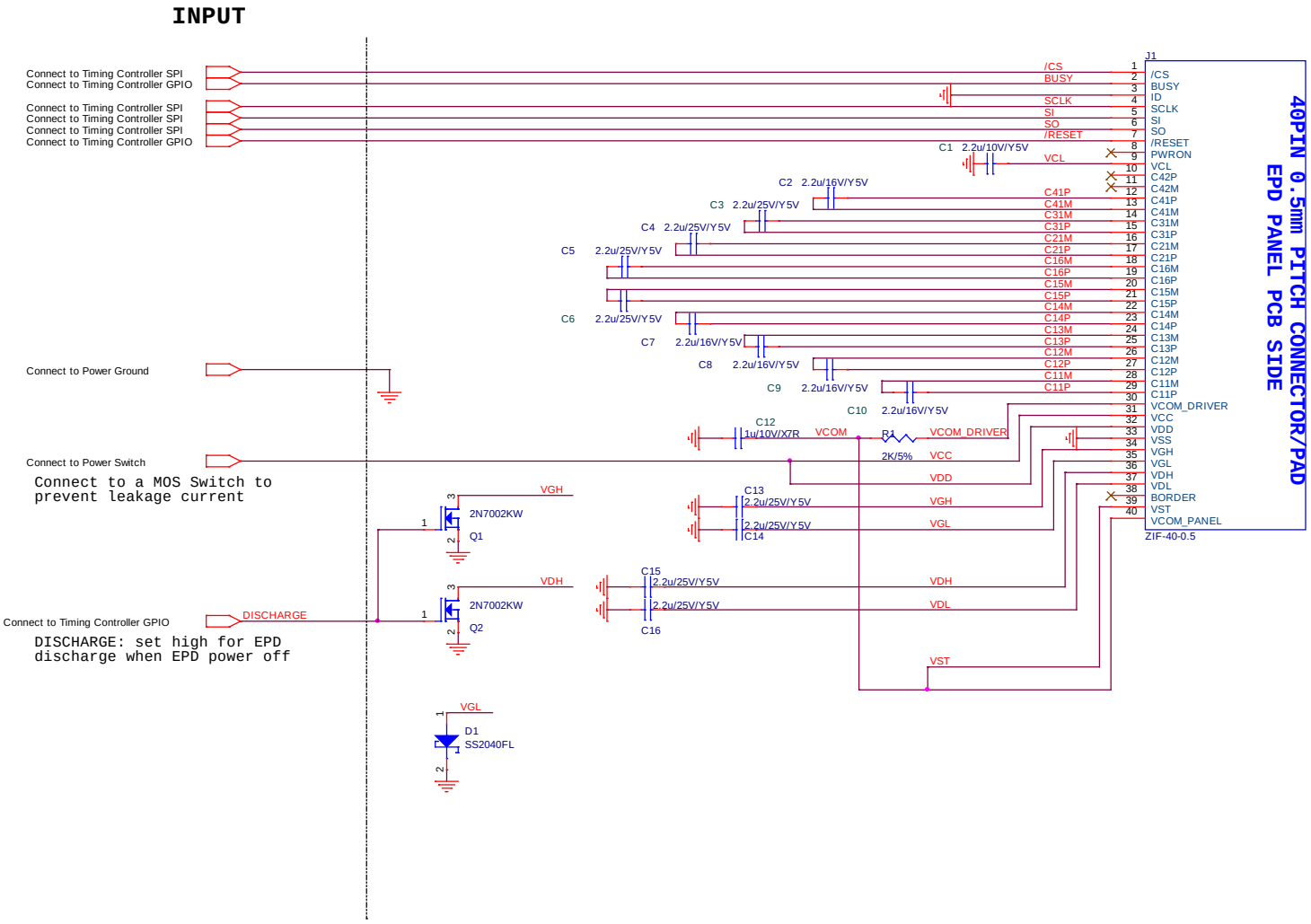


Note :

1. V_{DD} and V_{CC} must be discharged promptly after power off
2. 2.7" Pin.1 location

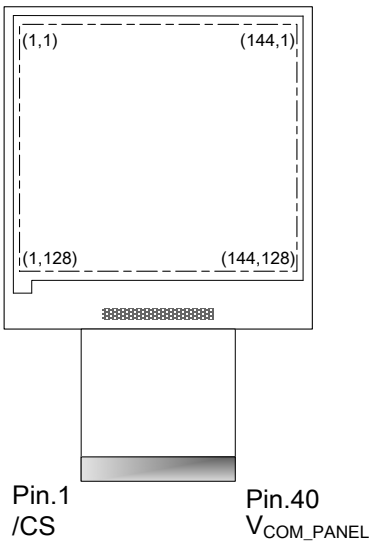


1.3.4 1.9 inch EPD Reference Circuit

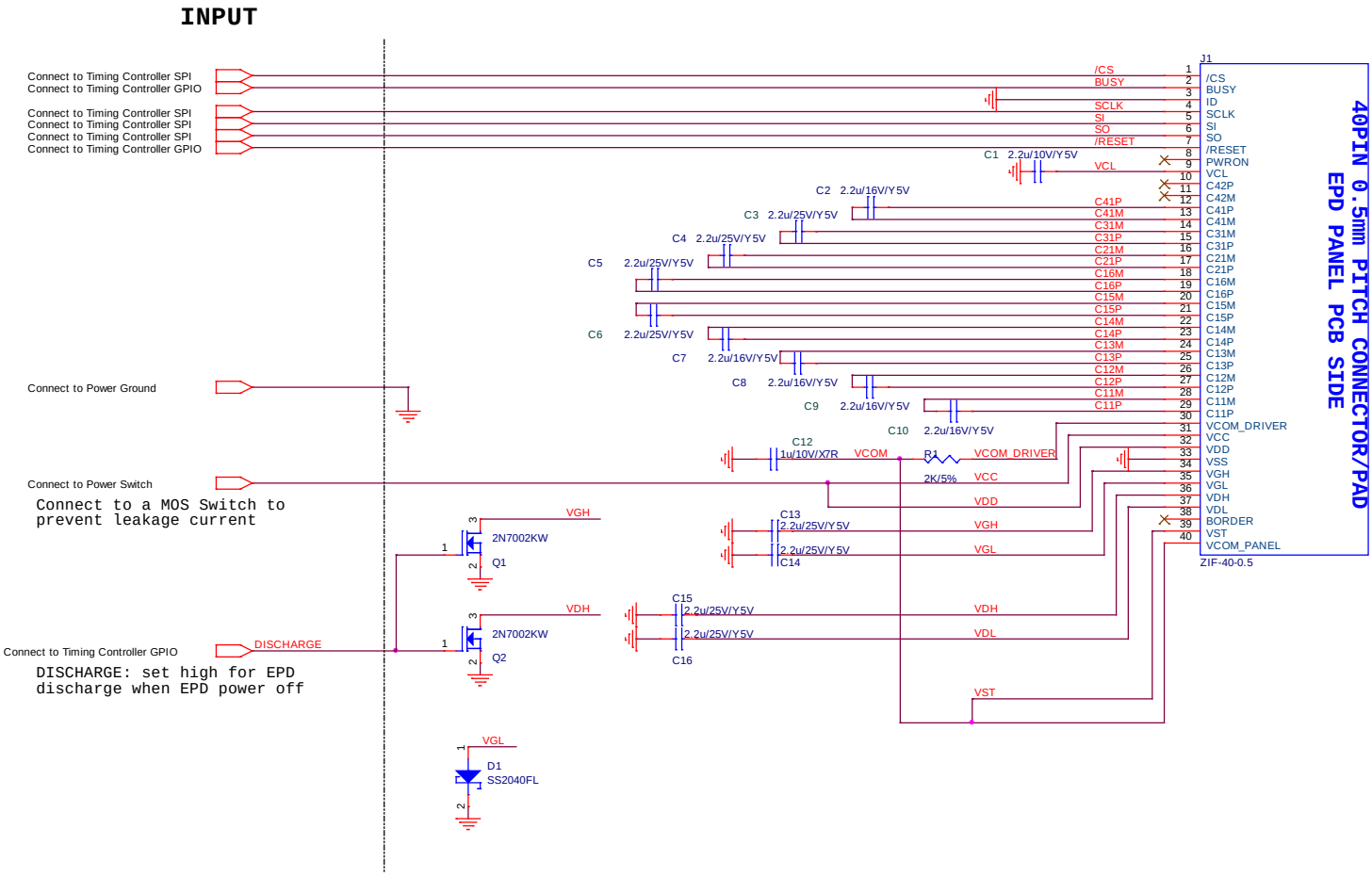


Note :

1. V_{DD} and V_{CC} must be discharged promptly after power off
2. 1.9" Pin.1 location

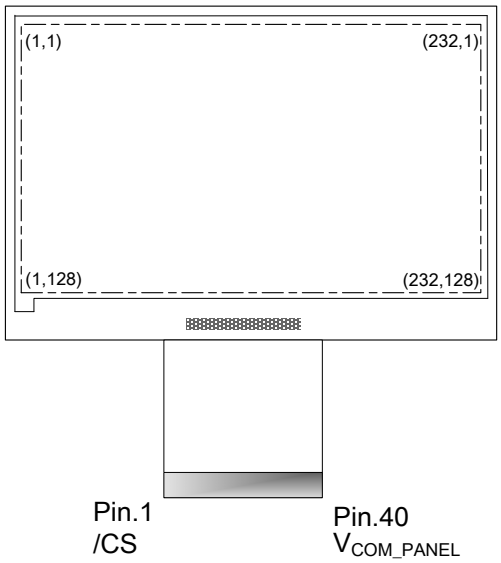


1.3.5 2.6 inch EPD Reference Circuit



Note :

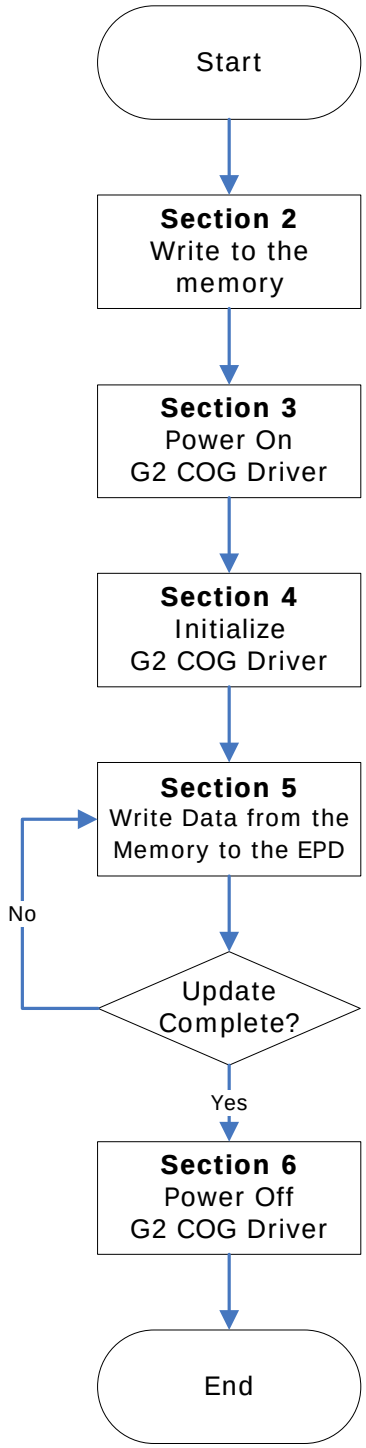
1. V_{DD} and V_{CC} must be discharged promptly after power off
2. 2.6" Pin.1 location



If you are still using our old COG driver G1 and previous circuit, please refer to the Appendix.

1.4 EPD Driving Flow Chart

The flowchart below provides an overview of the actions necessary to update the EPD. We call this is Global Update or Full Update. The steps below refer to the detailed descriptions in the respective sections.



1.5 Controller

The diagram below provides a signal control overview during an EPD update cycle.

The diagram is divided into

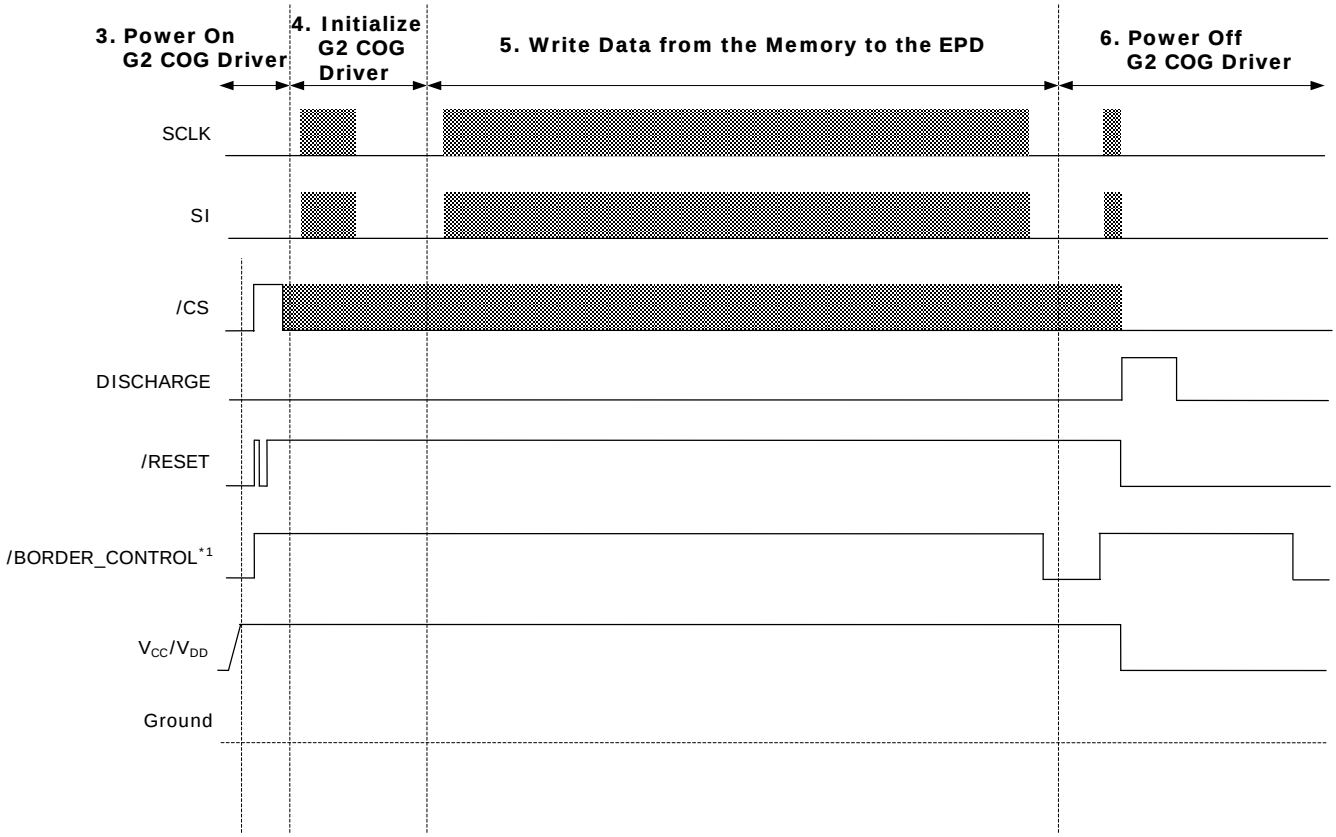
“3. Power On G2 COG Driver”,

“4. Initialize G2 COG Driver”,

“5. Write Data from the Memory to the EPD”,

“6. Power Off G2 COG Driver”,

The number and title matches a section title in this document.



Note:

1. /BORDER_CONTROL:
   /BORDER_CONTROL is used to keep a sharp border while taking care of the electronic ink particles. For implement this function, developer needs to use a Timing Controller pin (GPIO) to control this signal.

   (This function is only used for 2.7" EPD Panel)

1.6 SPI Timing Format

SPI commands are used to communicate between the Timing Controller and the G2 COG Driver. The SPI format differs from the standard in that two way communications, and /CS is pulled high then low between Command Index and Command Data. When setting up the SPI timing, please follow the SPI command format and timing in this section and verify the control signals in Section 1.5 then.

The maximum SPI clock speed for G2 COG Driver is 20MHz.

The SPI mode is 0.

Below is a description of the SPI Format:

SPI (0xI, 0xD_1, 0xD_2, 0xD_3 …)

Where:

I is the Command Index and the length is 1 byte
$D_{1\sim n}$ is the Command Data. The Command Data length is from 1 to 110 bytes depending on which Command Index is selected.

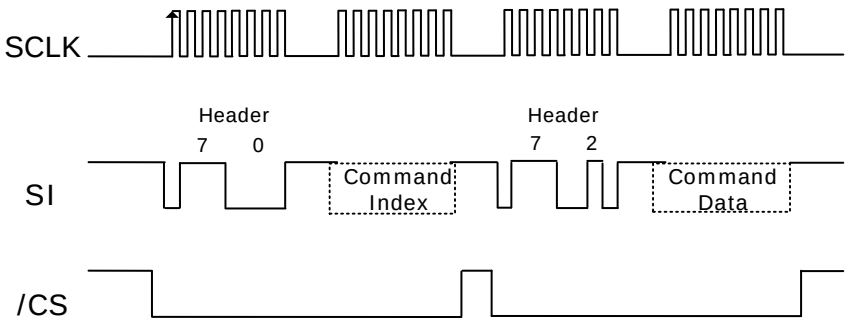
Command Index	Number Bytes of Command Data
0x01	8
0x02	1
0x03	1
0x04	1
0x05	1
0x07	1
0x08	1
0x09	1
0x0A	< 110
0x0B	1
0x0F	1

Before sending the Command Index, the SPI (SI) must send a 0x**70**(header of Command Index).

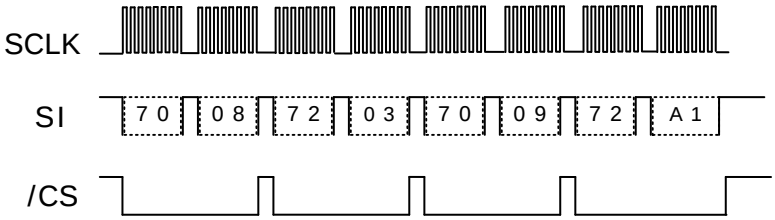
Likewise, the SPI (SI) must send a 0x**72**(header of Command Data) prior to the Command Data. The flow chart and detailed description can be found in the next page.

Number Bytes of Command Index (0x0A) depends on which panel size is used.

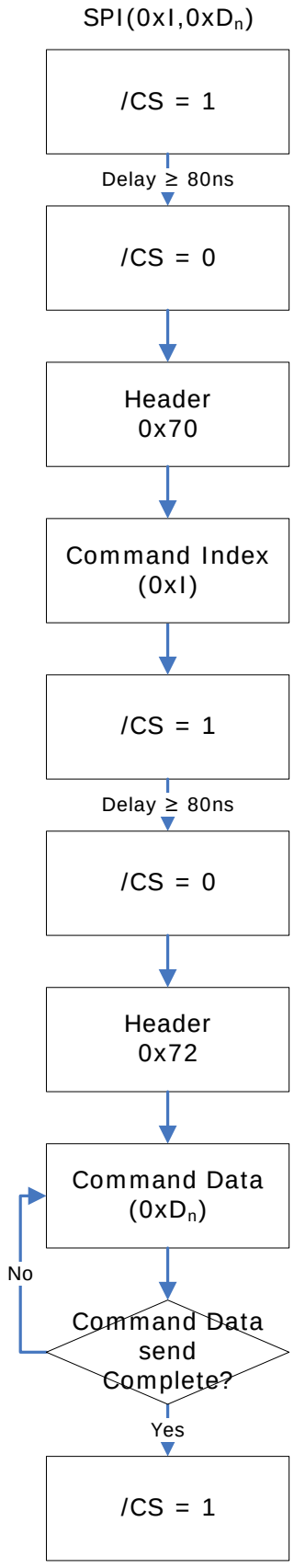
SPI write command signals and flowchart(SPI):



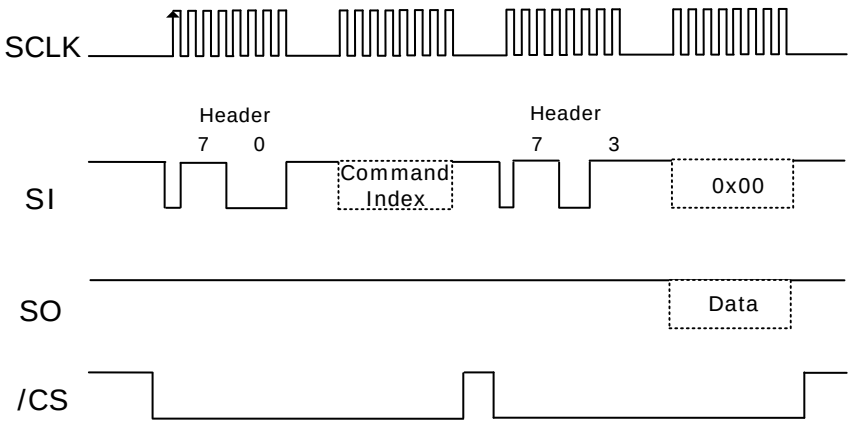
Example:
Send two continue SPI write commands
SPI(0x08,0x03) and SPI(0x09, 0xA1)



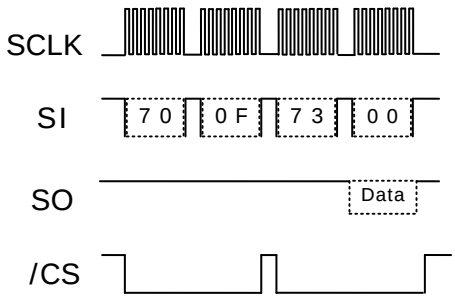
If Command Data is larger than two bytes, you must input data continuously without setting Command Index again.

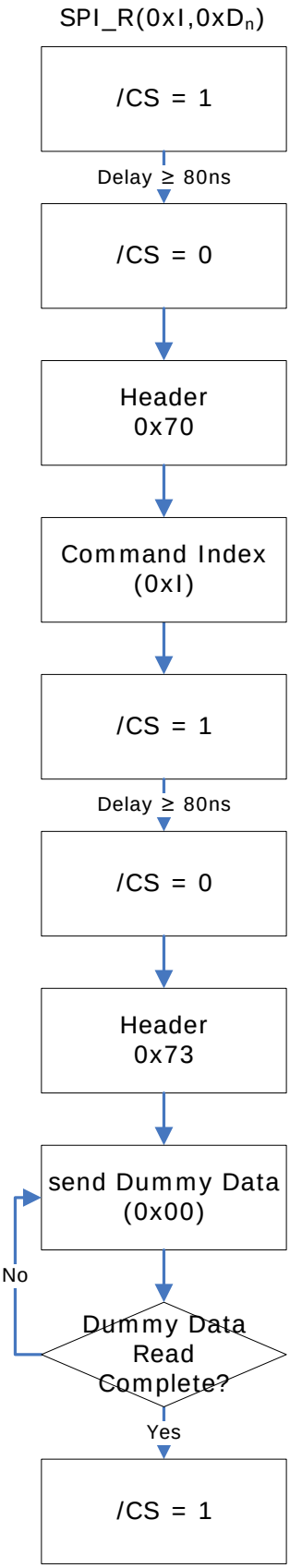


SPI read command signals and flowchart(SPI_R):

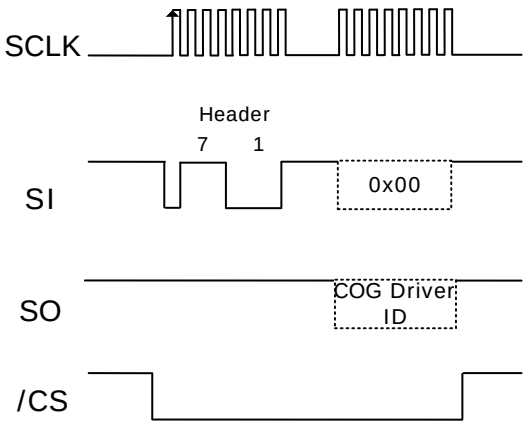


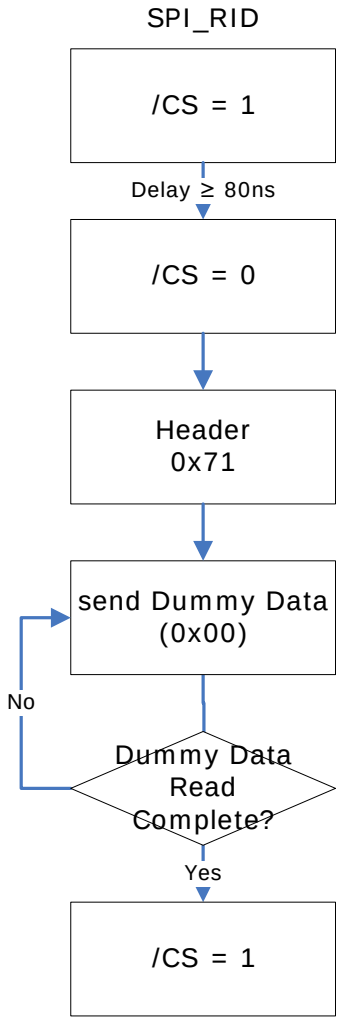
Example:
Send one SPI read command
SPI_R(0x0F,0x00)





SPI read COG Driver ID and flowchart(SPI_RID):





2. Write to the Memory

Before powering on G2 COG Driver, developer should write the new pattern to image buffer, either SRAM or flash memory. The image pattern must be converted to a 1 bit bitmap format (Black/White) prior to writing.

Two buffer spaces should be allocated to store both previous and new patterns. The previous pattern is the currently displayed pattern. The new pattern will be written to the EPD. The G2 COG Driver will compare both patterns before updating the EPD. The table below lists the buffer space size required for each EPD size.

EPD size	Image resolution(pixels)	Previous image + New image Buffer (bytes)
1.44”	128 x 96	3,072
2"	200 x 96	4,800
2.7”	264 x 176	11,616
1.9”	144 x 128	4,608
2.6”	232 x 128	7,424

3. Power On G2 COG Driver

This flowchart describes power on sequence for the G2 COG Driver.

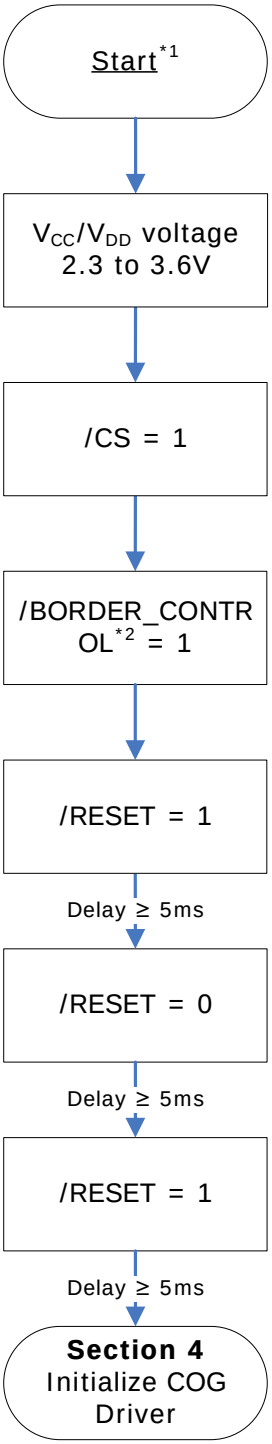
1. Start :

   Initial State:

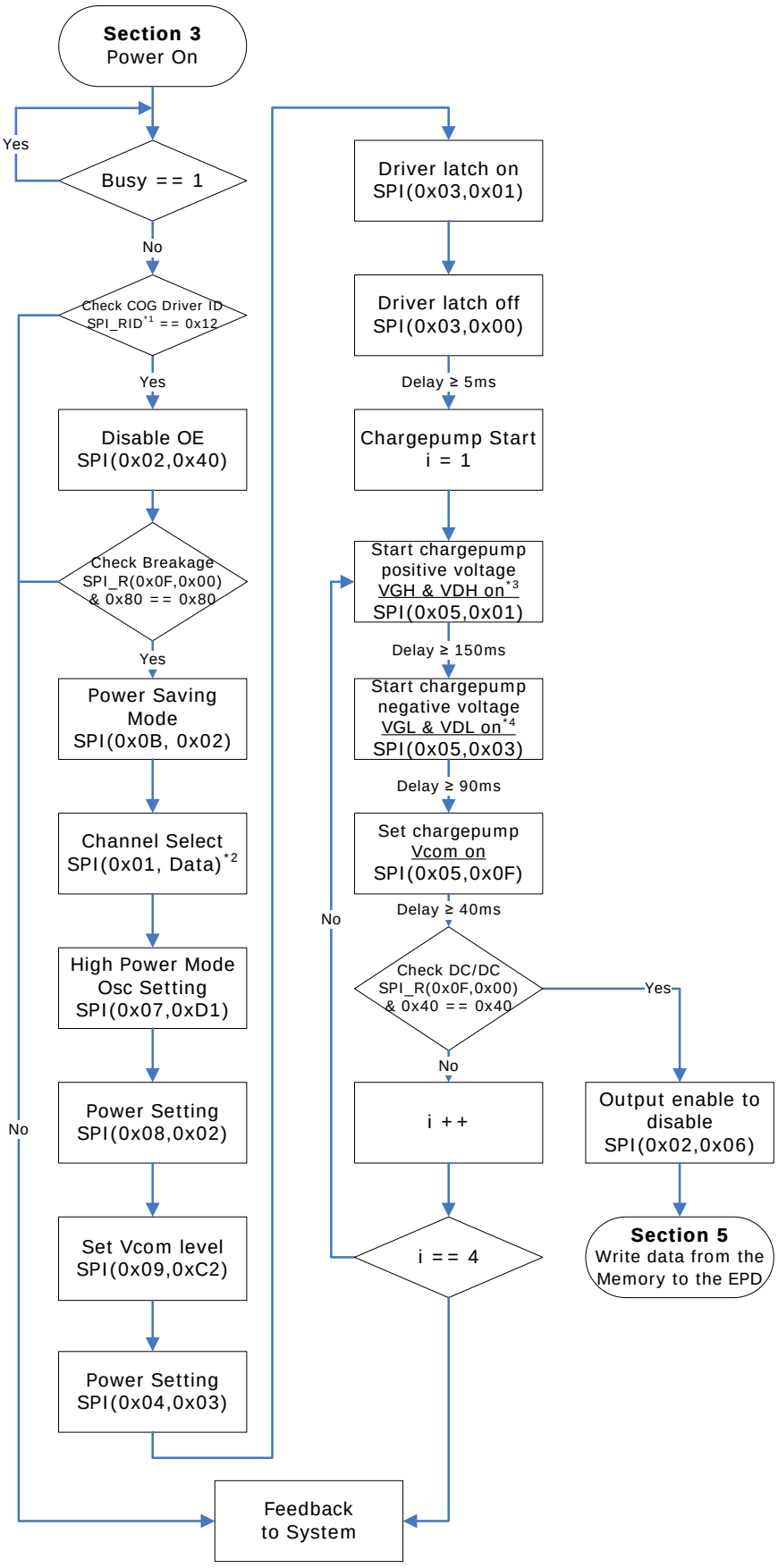
   V_{CC}/V_{DD} = 0

   /RESET, /CS, /BORDER_CONTROL[*2], SI, SCLK = 0

2. /BORDER_CONTROL:
   GPIO pin for 2.7" EPD Panel to reset the Border.



4. Initialize G2 COG Driver



Note:

1 SPI timing format (Refer to ”1.6 SPI Timing Format” for detail)

SPI read COG Driver ID command: SPI_RID

G1 COG Driver ID is 0x11

G2 COG Driver ID is 0x12

2 SPI(0x01, Data):

Different by each size

1.44”: SPI(0x01, (0x0000,0000,000F,FF00))

2”: SPI(0x01, (0x0000,0000,01FF,E000))

2.7”: SPI(0x01, (0x0000,007F,FFFE,0000))

1.9”: SPI(0x01, (0x0000,0003,FC00,00FF))

2.6”: SPI(0x01, (0x0000,1FE0,0000,00FF))

Take 2” for example, to send first byte protocol (0x70) before Command Index (0x01), and then send second byte protocol (0x72) before Command Data (0x0000,0000,01FF,E000).

3 Should measure VGH >12V and VDH >8V

4 Should measure VGL <-12V and VDL <-8V

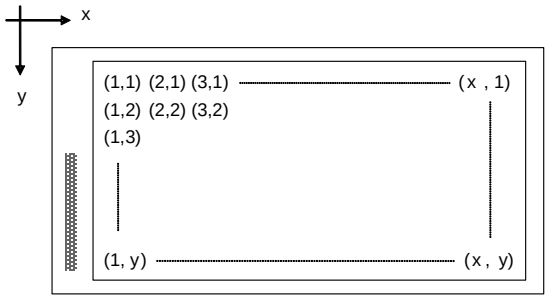
5 Write Data from the Memory to the EPD

5.1 Data Structure

This section describes how data should be sent to the G2 COG Driver which will update the display. The G2 COG Driver uses a buffer to store a line of data and then writes to the display.

EPD Resolutions

EPD size	Image resolution(pixels)	X	Y
1.44”	128 x 96	128	96
2"	200 x 96	200	96
2.7”	264 x 176	264	176
1.9”	144 x 128	144	128
2.6”	232 x 128	232	128



Data components

- One Bit – A bit can be W (White), B (Black) or N (Nothing) bit. Using the N bit to keep the pixel for the same color if new bit is same as previous one and also avoid overdriving the pixel to reduce ghosting phenomenon.
- One Dot/pixel is comprised of 2 bits.
- One Line is the number of dots in a line.

  For example:

  The 1.44” uses 128 Dots to represent 1 Line.

  The 2” uses 200 Dots to represent 1 Line.

  The 2.7” uses 264 Dots to represent 1 Line.

  The 1.9” uses 144 Dots to represent 1 Line.

  The 2.6” uses 232 Dots to represent 1 Line.

  The G2 COG Driver uses a buffer to write one line of data (Mapping) – interlaced. The table below is a Line data of 2" including Odd data bytes, Scan bytes and Even data bytes.

Data Bytes	Scan bytes	Data Bytes
1st – 25th (Odd)	1st - 24th	26th – 50th (Even)
{D(199,y),D(197,y), D(195, y), D(193, y)}…{D(7,y),D(5,y), D(3,y), D(1,y)}	{S(96),S(95), S(94),S(93)}…{S(4),S(3), S(2), S(1)}	{D(2,y),D(4,y), D(6,y), D(8,y)}…{D(194,y),D(196,y), D(198,y), D(200,y)}

- One frame of data is the number of lines * rows.

For example:

1.44": One frame of data is 96 lines * 128 dots.

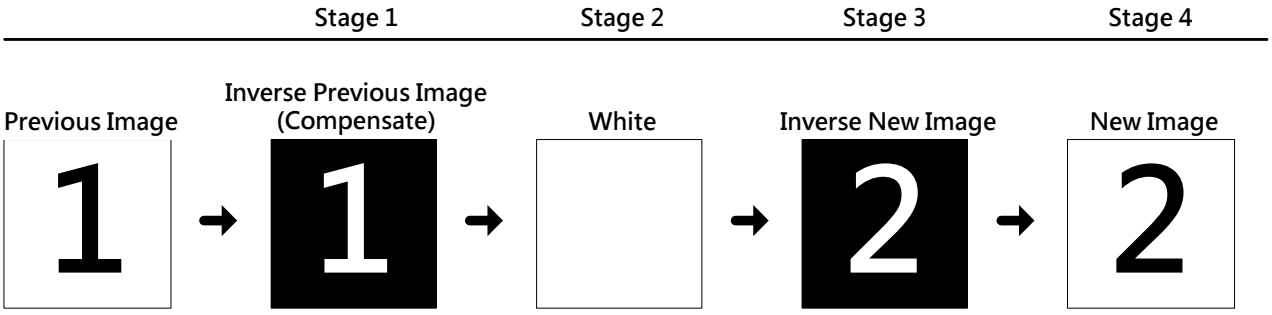
2": One frame of data is 96 lines * 200 dots.

2.7": One frame of data is 176 lines * 264 dots.

1.9": One frame of data is 128 lines * 144 dots.

2.6": One frame of data is 128 lines * 232 dots.

5.2 Overall Update Flow



5.3 Store a Line of Data in the Buffer

This section describes the details of how to send data to the G2 COG Driver. The G2 COG Driver uses a buffer to update the display line by line.

The Nothing data byte can be either 0x00 or 0x01 where 0x00 is the default Nothing data for G2 driver.

1.44” Input Data Order

Note :

1. Turn on OE :
   Output data from COG driver to panel.

2.

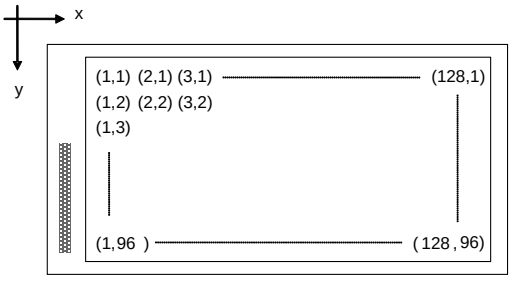
Data	bit1	bit0	Input
D(x,y) x = 1~128 y = 1~96	1	1	Black (B)
	1	0	White (W)
	0	0	Nothing (N)

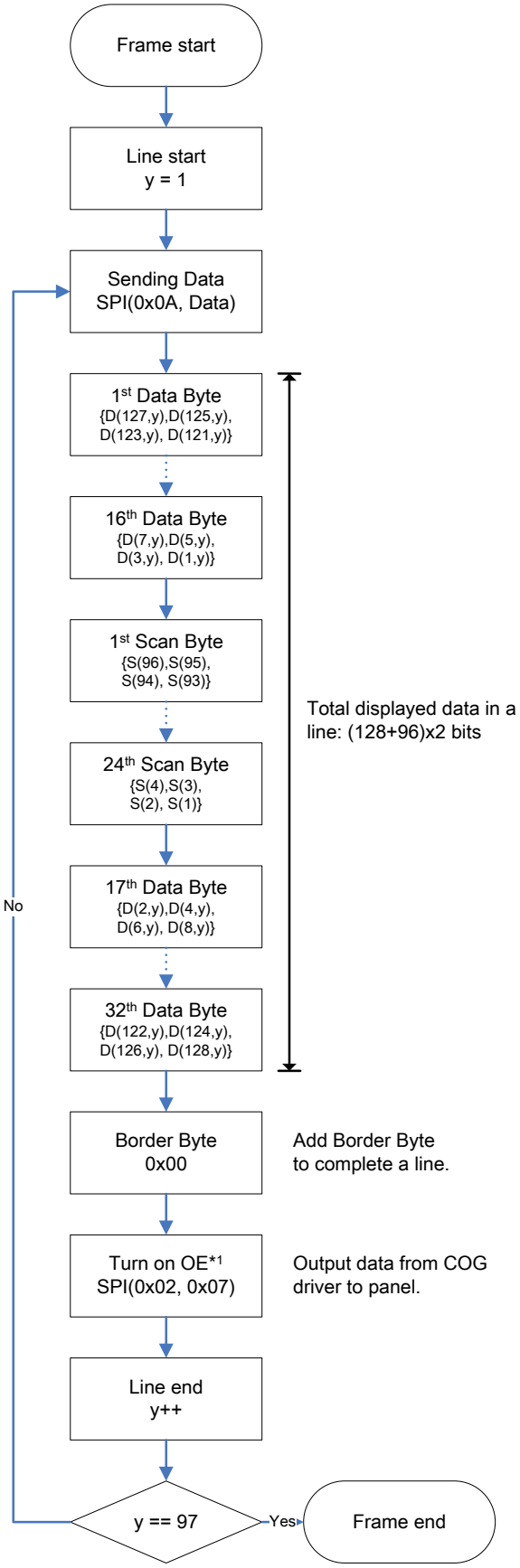
Example:
D(127,y) = Black (B) = 11
D(125,y) = White (W)= 10
D(123,y) = Nothing (N) = 00
D(121,y) = Black (B) = 11
1st Data Byte= 11,10,00,11

Scan	bit1	bit0	Input
S(1) ~S(96)	1	1	Scan on
	0	0	Scan off

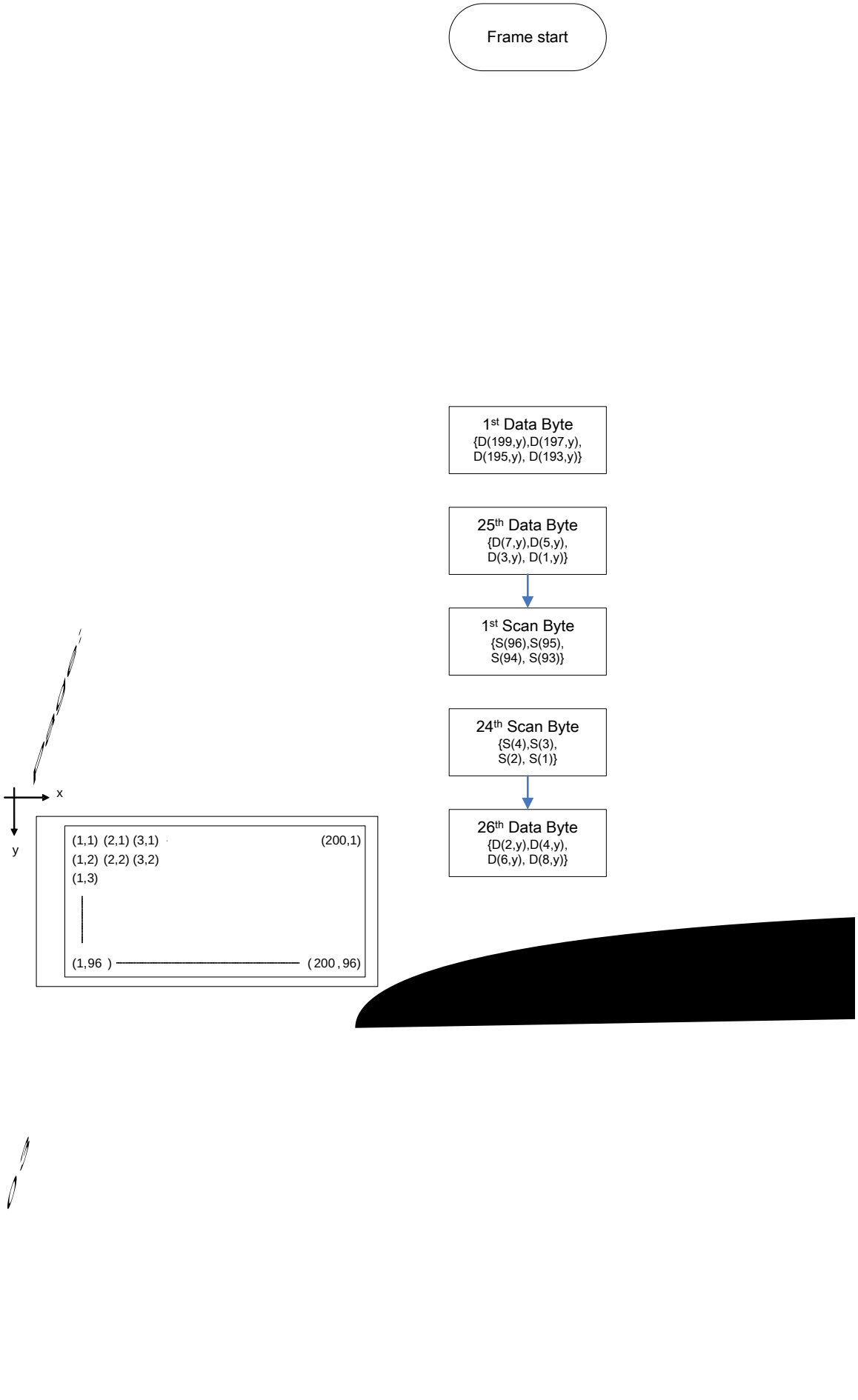
Example:
When y = 2,
Only S(2) is Scan on (11) while others are Scan off (00). The image represented by Data Bytes will be displayed on 2nd horizontal line (i.e. Dot(1,2) ~ Dot(128,2)).

S(1) = Scan off = 00
S(2) = Scan on = 11
S(3) = Scan off = 00
S(4) = Scan off = 00
:
S(96) = Scan off = 00
1st ~ 23rd Scan Byte = 00,00,00,00
24th Scan Byte = 00,00,11,00





2” Input Data Order



2.7” Input Data Order

Note :

1. Turn on OE :
   Output data from COG driver to panel.

2.

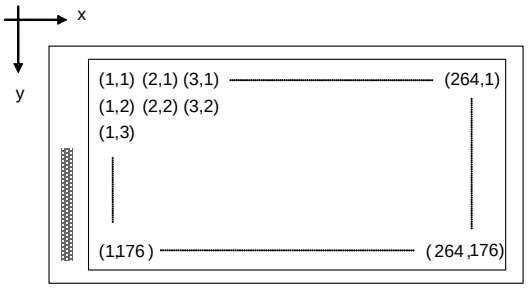
Data	bit1	bit0	Input
D(x,y) x = 1~264 y = 1~176	1	1	Black (B)
	1	0	White (W)
	0	0	Nothing (N)

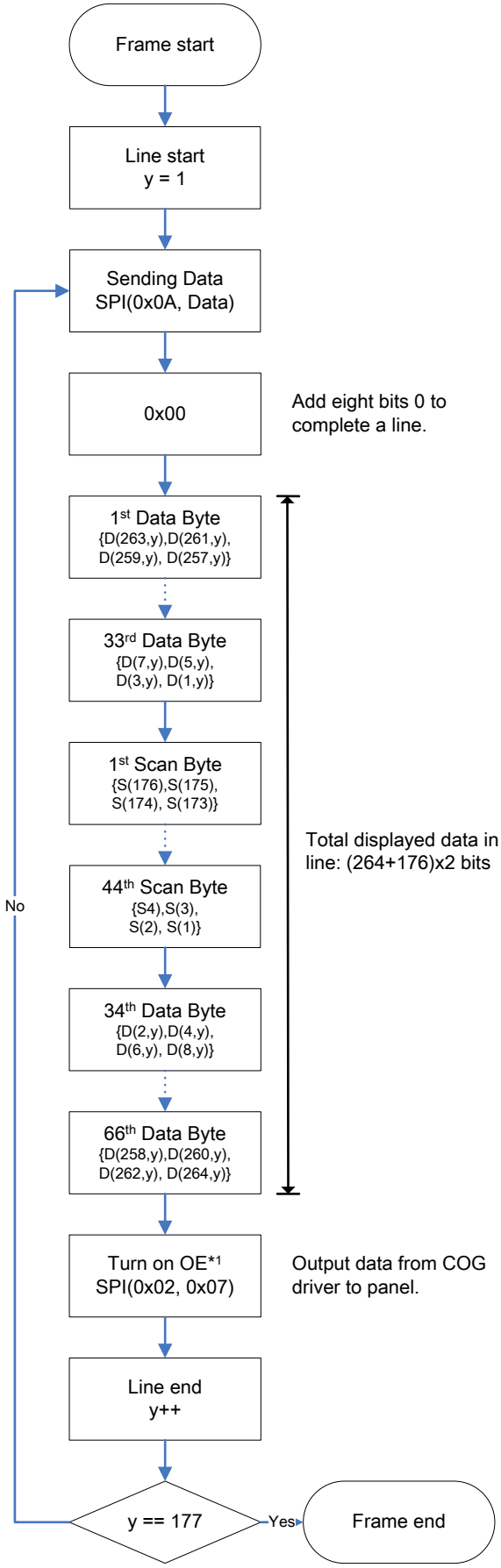
Example:
D(263,y) = Black (B) = 11
D(261,y) = White (W)= 10
D(259,y) = Nothing(N) = 00
D(257,y) = Black (B) = 11
1st Data Byte= 11,10,00,11

Scan	bit1	bit0	Input
S(1) ~S(176)	1	1	Scan on
	0	0	Scan off

Example:
When y = 2,
Only S(2) is Scan on (11) while others are Scan off (00). The image represented by Data Bytes will be displayed on 2nd horizontal line (i.e. Dot(1,2) ~ Dot(264,2)).

S(1) = Scan off = 00
S(2) = Scan on = 11
S(3) = Scan off = 00
S(4) = Scan off = 00
:
S(176) = Scan off = 00
1st ~ 43rd Scan Byte = 00,00,00,00
44th Scan Byte = 00,00,11,00





1.9” Input Data Order

Note :

1. Turn on OE :
   Output data from COG driver to panel.

2.

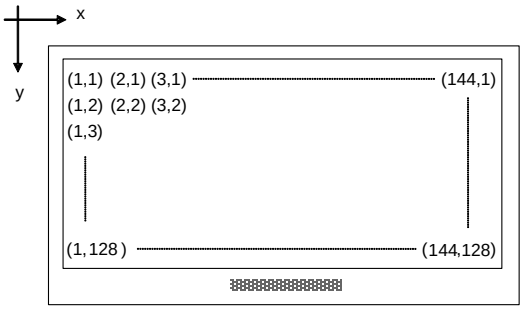
Data	bit1	bit0	Input	
D(x,y) x = 1~144 y = 1~128	1	1	Black	(B)
	1	0	White	(W)
	0	0	Nothing	(N)

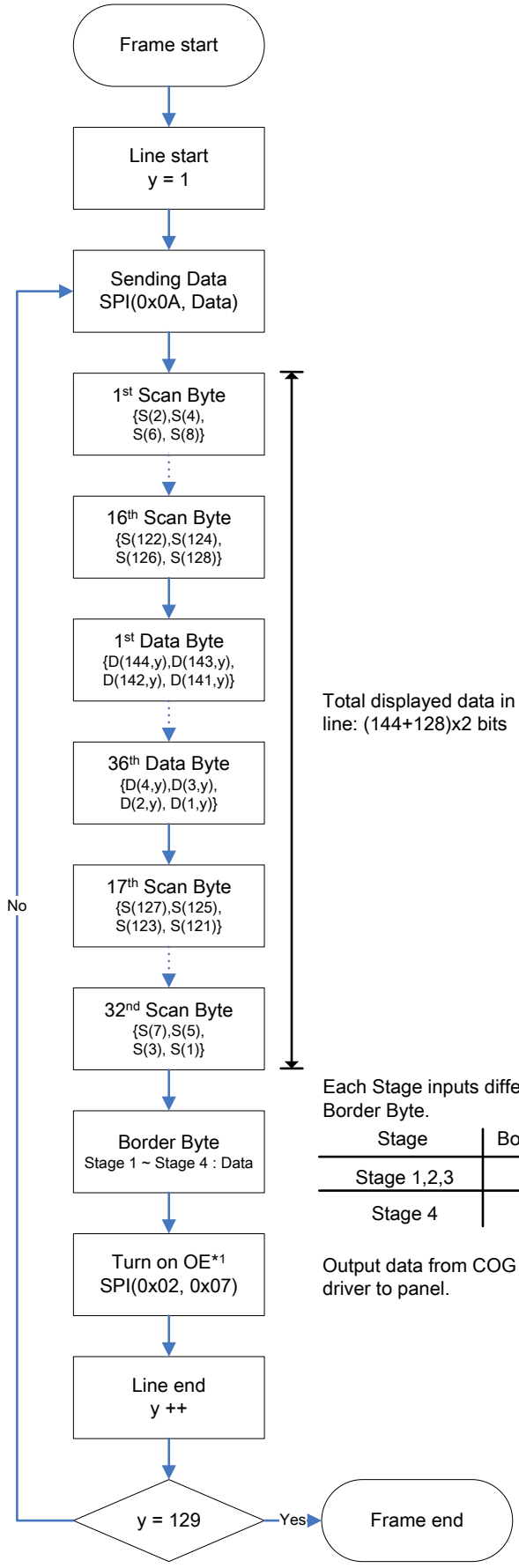
Example:
D(1,y) = Black (B) = 11
D(2,y) = White (W)= 10
D(3,y) = Nothing (N) = 00
D(4,y) = Black (B) = 11
1st Data Byte= 11,10,00,11

Scan	bit1	bit0	Input
S(1) ~S(128)	1	1	Scan on
	0	0	Scan off

Example:
When y = 2,
Only S(2) is Scan on (11) while others are Scan off (00). The image represented by Data Bytes will be displayed on 2nd horizontal line (i.e. Dot(1,2) ~ Dot(144,2)).

S(1) = Scan off = 00
S(2) = Scan off = 11
S(3) = Scan off = 00
S(4) = Scan off = 00
S(5) = Scan off = 00
S(6) = Scan off = 00
S(7) = Scan off = 00
S(8) = Scan off = 00
:
S(128) = Scan off = 00
1st Scan Byte = 11,00,00,00
2nd ~ 32nd Scan Byte = 00,00,00,00





Total displayed data in a line: (144+128)x2 bits

Each Stage inputs different Border Byte.

Stage	Border Byte
Stage 1,2,3	0x00
Stage 4	0xAA

Output data from COG driver to panel.

2.6” Input Data Order

Note :

1. Turn on OE :
   Output data from COG driver to panel.

2.

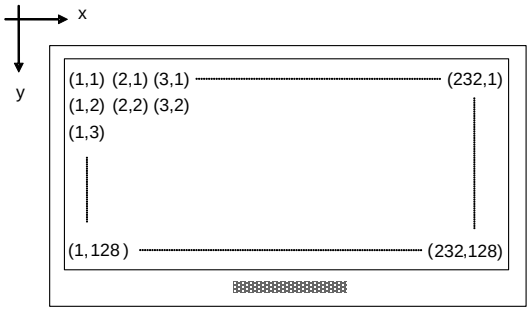
Data	bit1	bit0	Input
D(x,y) x = 1~232 y = 1~128	1	1	Black (B)
	1	0	White (W)
	0	0	Nothing (N)

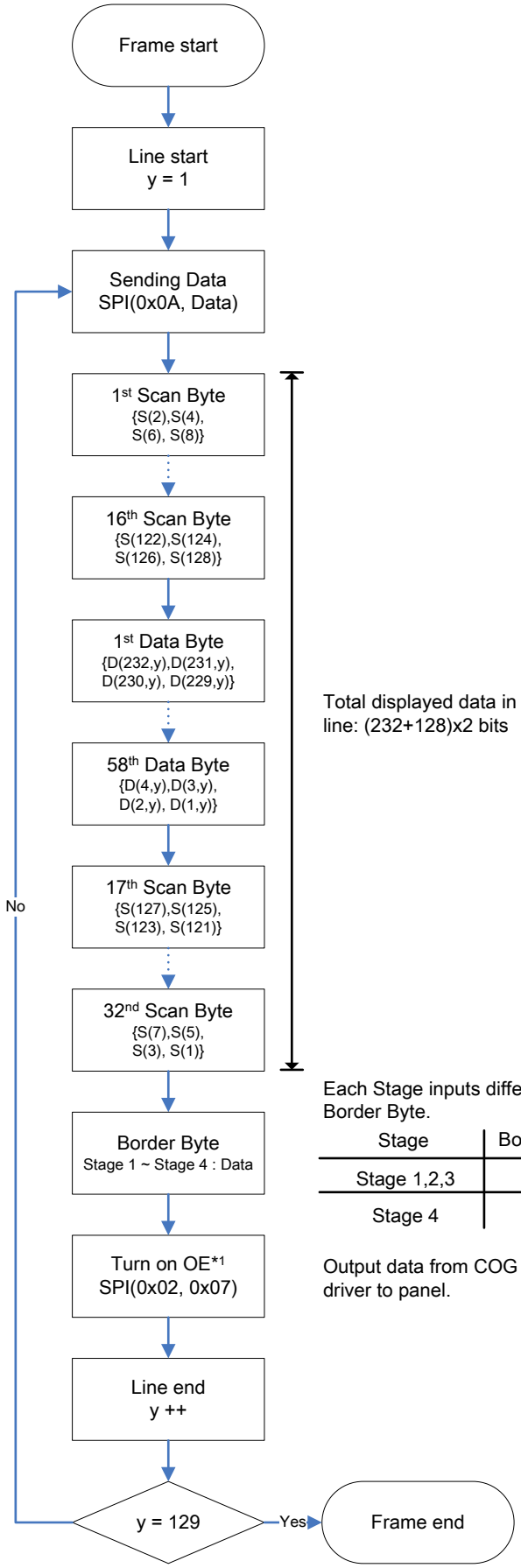
Example:
D(1,y) = Black (B) = 11
D(2,y) = White (W)= 10
D(3,y) = Nothing (N) = 00
D(4,y) = Black (B) = 11
1st Data Byte= 11,10,00,11

Scan	bit1	bit0	Input
S(1) ~S(128)	1	1	Scan on
	0	0	Scan off

Example:
When y = 2,
Only S(2) is Scan on (11) while others are Scan off (00). The image represented by Data Bytes will be displayed on 2nd horizontal line (i.e. Dot(1,2) ~ Dot(232,2)).

S(1) = Scan off = 00
S(2) = Scan off = 11
S(3) = Scan off = 00
S(4) = Scan off = 00
S(5) = Scan off = 00
S(6) = Scan off = 00
S(7) = Scan off = 00
S(8) = Scan off = 00
:
S(128) = Scan off = 00
1st Scan Byte = 11,00,00,00
2nd ~ 32nd Scan Byte = 00,00,00,00

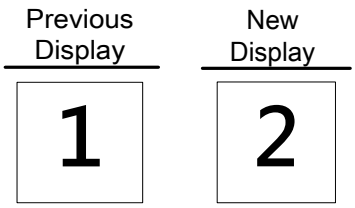




5.4 Writing to the Display in Stages

This section contains the method to write to the display in stages. Each of the 4 stages should be the same use the same number of frames. Rewrite the frame during each stage.

The following flow chart describes how to update an image from a previous displayed image stored in memory buffer to a new image also stored in memory buffer. See the sample previous and new images below.



Panel Size	Stage Time (ms)*1	Frame Time of Timing Controller (ms)*2
1.44”	480	< 50ms
2”	480	< 50ms
1.9”	480	< 50ms
2.7”	630	< 80ms
2.6”	630	< 70ms

Note:

1 Stage Time is the number of frames used to write an intermediate pattern. This can vary based on the Frame Time of Timing Controller (or MCU).

Example: Use 2 inch for example

If the Frame Time of Timing Controller is 40ms, it needs 12 frames at least of data per stage. (480/40 = 12)

If the Frame Time of Timing Controller is 36ms, it cannot divisible the Stage Time, users can choose one of the two methods as below to determine the number of frames

(1) Frame Time = 36ms (480/36 = 13.33), users can drive 14 frames of data per stage at least.

(2) Frame Time = 36ms (480/36 = 13.33), users can drive 13 frames of data per stage. And use equal delay time in each frame to meets the Stage Time.

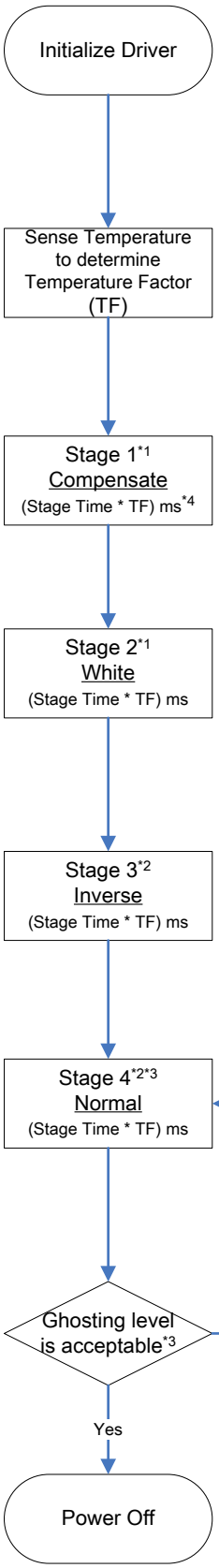
Example: 36ms * 13 + delay time= 480ms.

2 Frame Time is based on the MCU that PDi used to recommend a maximum limit value of Frame time.

3 This table is tested with PDi’s jig.

Temperature (℃)	TF*5
≤-10	17
-5 ≥ T > -10	12
5 ≥ T > -5	8
10 ≥ T > 5	4
15 ≥ T > 10	3
20 ≥ T > 15	2
40 ≥ T > 20	1
> 40	0.7

1. The previous image stored in memory is used to determine how to write the data for both Stage 1 and Stage 2.
2. The new image stored in memory is used to determine how to write the data for both Stage 3 and Stage 4.
3. Optional: The optical performance is dependent on Stage Time. If the ghosting is at unacceptable level, the EPD can be rewritten and then Stage 4 repeated to write the New image.
4. It needs (Stage Time * TF) ms to finish a stage.
5. The TF below 0℃ is for reference only. PDI does not guarantee the performance and functionality below 0℃.
6. If you use Flash memory for the Section 2, please erase the buffer when Stage 4 is completed.



	bit1	bit0	Input	Previous Display
Data	1	1	Black (B)	1
	1	0	White (W)	
	0	0	Nothing (N)	

Stage 1	Data		Display
Previous*1	B	W	1
Input	W	B	
Display	W	B	

Stage 2	Data		Display
Previous*1	B	W	White
Input	N	W	
Display	W	W	

Stage 3	Data		Display
New*2	B	W	2
Input	N	B	
Display	W	B	

Stage 4	Data		New Display
New*2	B	W	2
Input	B	W	
Display	B	W	

Stage R	Data		Display
New*2	B	W	2
Input	W	B	
Display	W	B	

6 Power Off G2 COG Driver

1. Nothing Frame :
   Write a frame data

Panel size	Scan Line
1.44"	96
2"	96
2.7"	176
1.9"	128
2.6"	128

   Whose Data Bytes are 0x00. Scan Bytes operate normally.
   Scan lines are still turned on sequentially. This frame will make the image more uniform. Turn on OE SPI(0x02, 0x07) at the end of each line.
   For 1.44", 1.9", 2" & 2.6", need to set Border Byte to complete 1st Data Byte.
   Border Byte = 0x00 for 1.44" & 2"
   0xAA for 1.9" & 2.6"

2. Dummy Line :
   A line whose all Data Bytes are 0x00 and Scan Bytes are 0x00. Turning on OE SPI(0x02, 0x07) to complete this Dummy Line. Clear the register data before power off. Detail of data input is on page 27 ~ page 31.
   (This function is only used in 2.7")

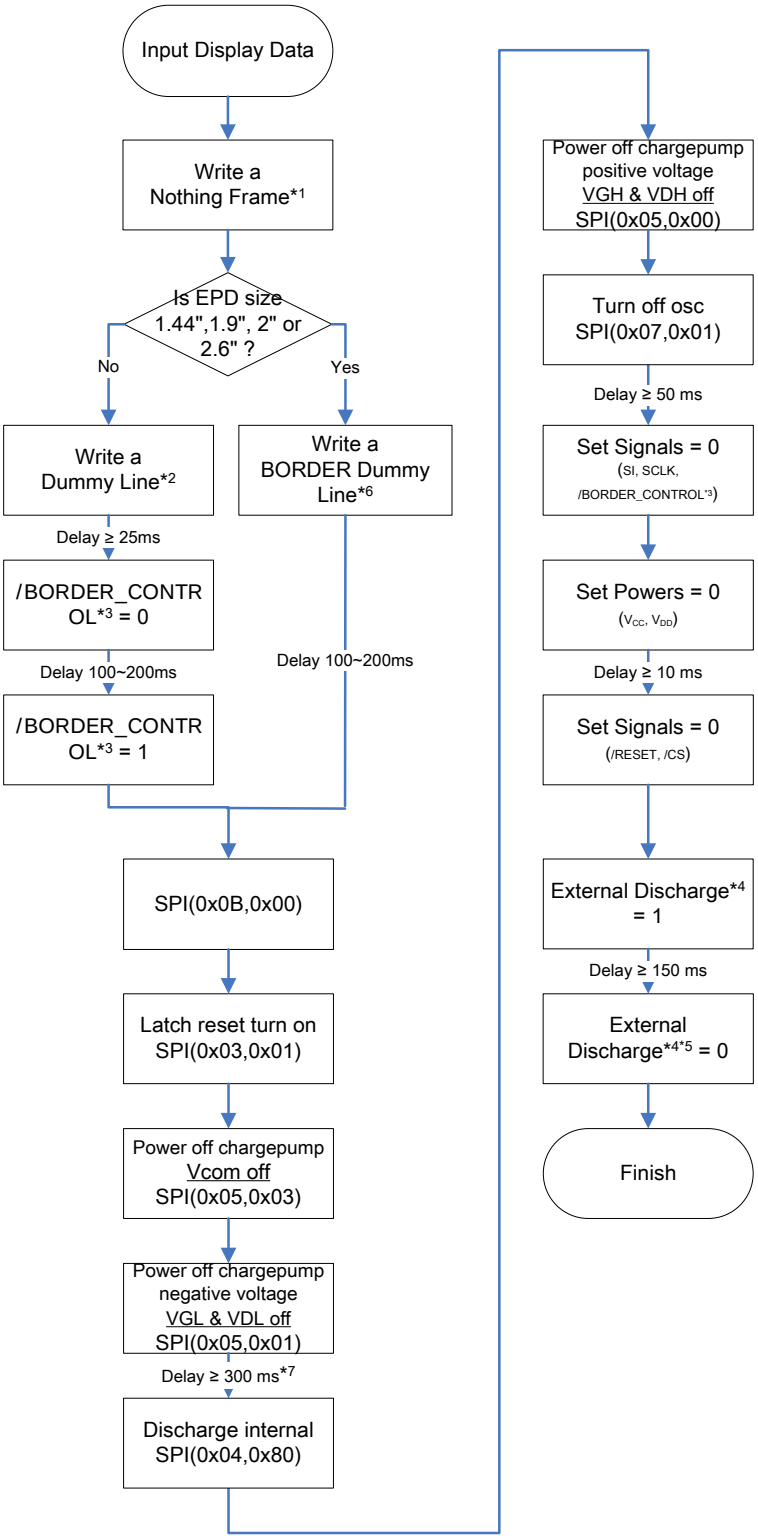
3. /BORDER_CONTROL :
   When = 0, the /BORDER_CONTROL is ON and write to white. When = 1,the /BORDER_CONTROL is OFF.
   (This function is only used in 2.7")

4. External Discharge :
   For implement this function, users need to use a DISCHARGE pin from microcontroller to control.(refer to the reference circuit)
   This is important to avoid vertical lines.

5. If you use the flash memory for pattern store, please recheck flash in this phase and verify the old image flash is erased.

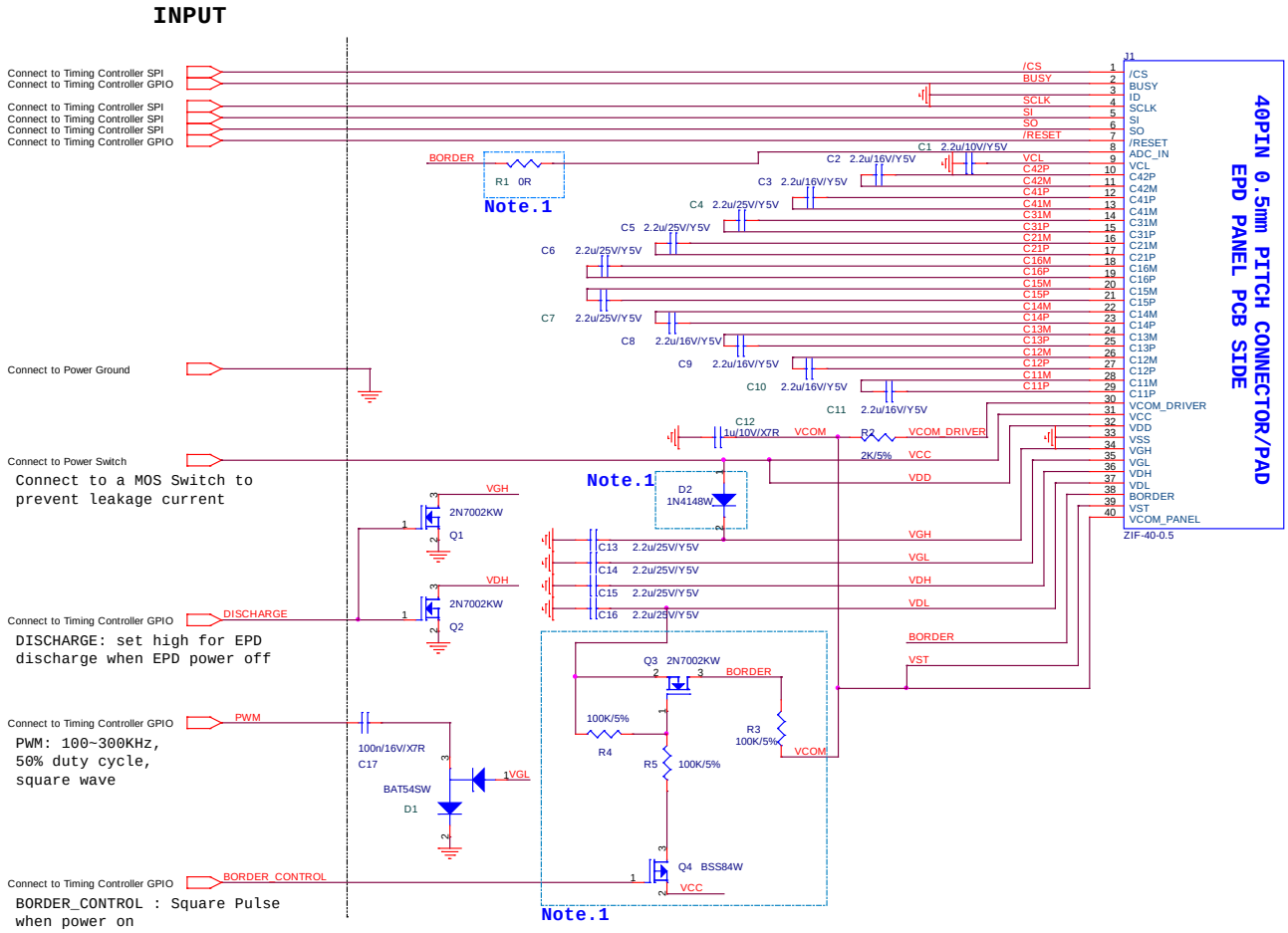
6. BORDER Dummy Line :
   Set Border Byte = 0xAA and write to white. A line whose all Data Bytes are 0x00 and Scan Bytes are 0x00. Then must to set SPI(0x02,0x07) in the end of line for turn on output enable of COG Driver to control border and clear the register data before power off. Detail of data input is on page 27 ~ page 31.
   (This function is only used in 1.44" ,1.9" ,2" & 2.6")

7. Delay time of VGL and VDL :
   Please ensure to place the accurate delay time here until VGL is discharged to GND by measured by oscilloscope.



Appendix: Use G1's PCBA to drive the EPD with G2 Driver IC

Below is the reference circuit if you have arranged the PCBA that drove the EPD with G1 Driver IC already.



Note:

1. Hardware setting for different size:

	R1	Q3,Q4,R3,R4,R5	D2
1.44 inch & 2 inch	Mounted	No Mounted	Mounted
2.7 inch	No Mounted	Mounted	No Mounted
1.9 inch & 2.6 inch	No Mounted	No Mounted	No Mounted

If users want to drive the EPD with G2 Driver IC by the current PCBA (i.e. the reference circuit above). Below items are the steps needed to do.

Keep hardware unchanged as above.

Keep Resistor R1 open.

Keep BORDER CONTROL (Q3, Q4, R3, R4, and R5) circuit mounted.

Modify SPI data as the following sections described.

Disable the Timing Controller GPIO pin, PWM. Keep PWM signal as either 1 or 0.

No matter what size EPD is, use same power off sequence of section 6.