DIGITAL MULTIPHASE

GEN 1 PROGRAMMING GUIDE

JANUARY 2020

BIG IDEAS
FOR EVERY SPACE

RENESAS

OVERVIEW

- **This guide specifies the algorithm for programming Renesas digital multiphase products via PMBus communication.**

- **Unless noted otherwise, timing and voltage requirements are outlined in the PMBus specification version 1.3.**

- **The following sections are shown in this guide:**
  - Section 1: Programming new devices
  - Section 2: Reprogramming devices

Supported Devices

Device Name	Package	IC_DEVICE_ID Byte ID (Step 2b)	Hex Version (Step 2a)
ISL68124	40 LD QFN	0x1E	0x05040000
ISL68127	48 LD QFN	0x28	0x07070000
ISL68134	40 LD QFN	0x1F	0x05040000
ISL68137	48 LD QFN	0x27	0x07070000
ISL68144	40 LD QFN	0x22	0x05040000
ISL68147	48 LD QFN	0x29	0x07070000
ISL69122	40 LD QFN	0x2E	0x05040000
ISL69124	40 LD QFN	0x2B	0x05040000
ISL69125	40 LD QFN	0x2C	0x05040000
ISL69127	48 LD QFN	0x23	0x07070000
ISL69128	48 LD QFN	0x24	0x07070000
ISL69129	48 LD QFN	0x2A	0x07070000
ISL69133	40 LD QFN	0x30	0x05040000
ISL69134	40 LD QFN	0x2D	0x05040000
ISL69137	48 LD QFN	0x25	0x07070000
ISL69138	48 LD QFN	0x31	0x07070000
ISL69144	40 LD QFN	0x2F	0x05040000
ISL69147	48 LD QFN	0x26	0x07070000
ISL69158	48 LD QFN	0x32	0x05040000

This programming guide supports the devices shown in the Device Table above.

Prerequisites and Conventions

- HEX configuration files must be generated using PowerNavigator.
- The address specified within the HEX file must match the PMBus address used in the programming socket or on the board.
- In this guide, address 0x60 (7-bit format) is used in all examples.
- Data on the bus may be reversed. Follow examples for correct byte order.

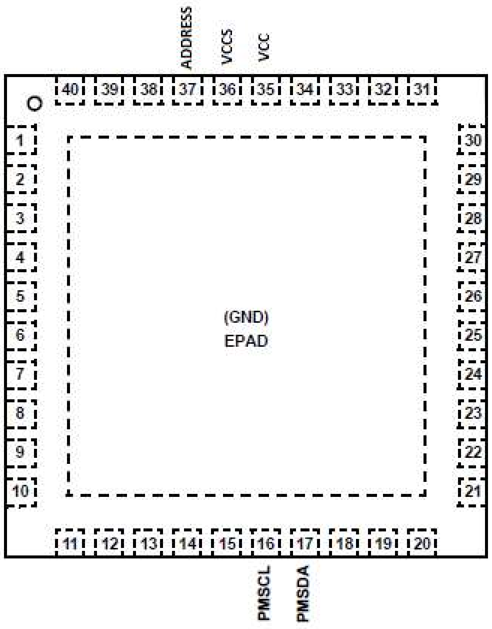
Minimum Pin and Component Requirements

The following pins must be connected when programming a device:

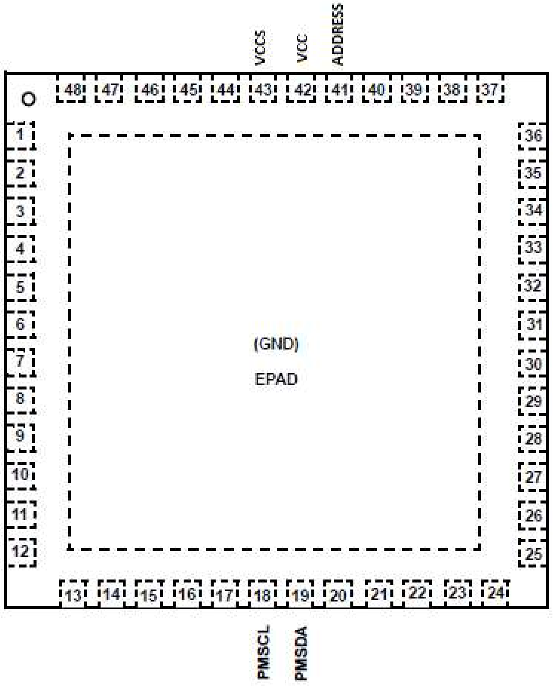
- PMSCL and PMSDA (I^2C clock and data pins). These are open drain and must be pulled to 3.3V via a resistor (1kΩ maximum).
- VCC, provided with an external 3.3V supply. A 1uF decoupling capacitor from this pin to ground is also needed.
- VCCS must be decoupled with 4.7µF or greater MLCC (X5R or better).
- Ground pin must be connected to ground.
- ADDRESS pin must have an address set resistor to ground. Connect directly to ground for address 0x60.
- Other pins may be floated.

Minimum Pin and Component Requirements

40 LD QFN
TOP VIEW

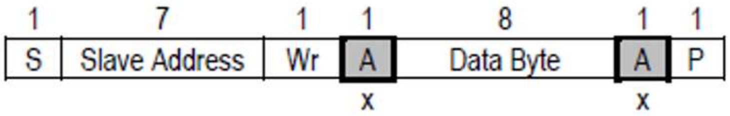


(48 LD QFN)
TOP VIEW



- VCC = 3.3V, 1uF MLCC decoupling cap to GND.
- VCCS = 4.7uF MLCC decoupling cap to GND.
- PMSCL = 4.7k to 1k pull-up to VCC
- PMSDA = 4.7k to 1k pull-up to VCC
- ADDRESS = Address set resistor (GND for address 0x60)

PMBus Communication Key



1	7	1	1	8	1	1
S	Slave Address	Wr	A	Data Byte	A	P
			x		x	

S	Start Condition
Sr	Repeated Start Condition
Rd	Read (bit value of 1)
Wr	Write (bit value of 0)
x	Shown under a field indicates that that field is required to have the value of ‘x’
A	Acknowledge (this bit position may be ‘0’ for an ACK or ‘1’ for a NACK)
P	Stop Condition
PEC	Packet Error Code
□	Master-to-Slave
▨	Slave-to-Master
...	Continuation of protocol

Note: See PMBus/SMBus spec for additional details and timing requirements.

Direct Memory Access (DMA) Command Codes

Actual device programming is completed through 3 command codes:

- **DMA Address (Command Code 0xF7):** Used to set the register address to use with other DMA commands.
- **DMA Data (Command Code 0xF5):** Used to read from or write to the register selected by the DMA Address command.
- **DMA Sequential (Command Code 0xF6):** Used to read from or write to the register selected by the DMA Address command, then automatically increment the register address by 1.

Important Notes

- Once the first write from the HEX file to the part has occurred, the part is in programming mode. Regardless of whether programming completes or is suspended, controller VCC must be cycled to clear this mode and put the part back in normal processing mode.

- The command that causes the OTP to burn is the last line of the HEX file. Device programming can be aborted at any point before this, and no contents will be burned to OTP.

- The last byte of each line in the HEX file is a CRC8 check for that line in the file only. It does not relate to any hardware value.

SECTION 1:

PROGRAMMING NEW DEVICES

PROGRAMMING ALGORITHM OVERVIEW

1. **Determine number of NVM slots available. (Optional)**

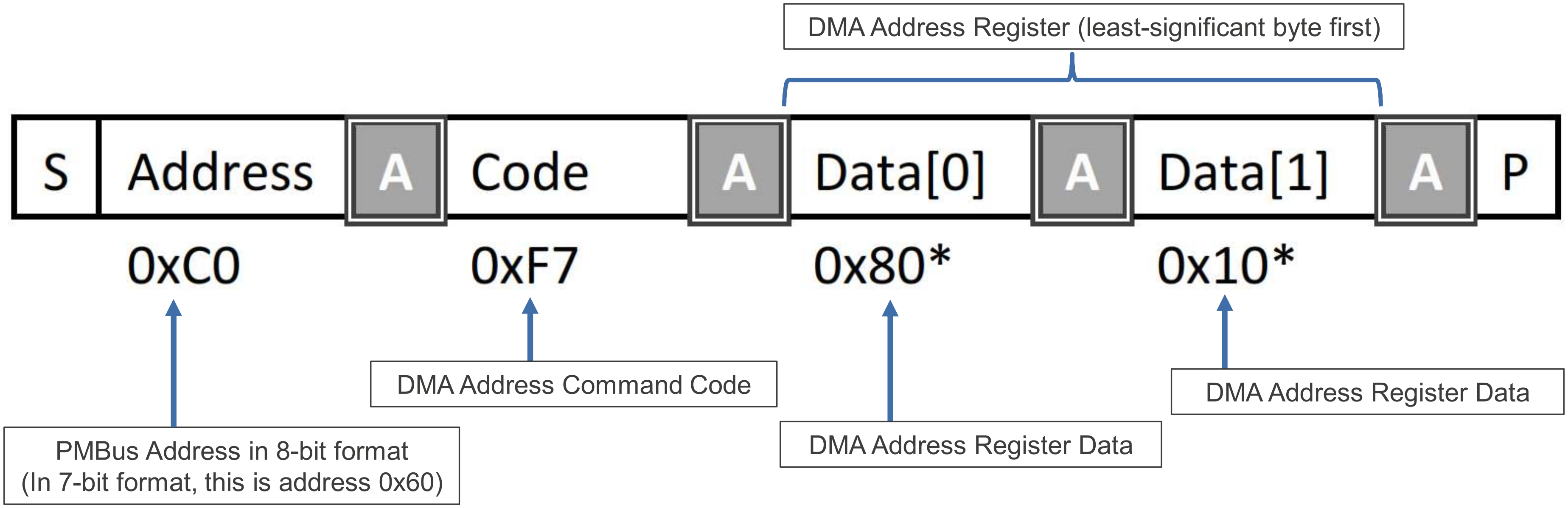
2. **Verify device and file versions.**
   a. Read and parse header data from HEX file. Go to the Step 2a section for more information.
   b. Read IC_DEVICE_ID from device. Verify the value matches the Device Table.
   c. Read IC_DEVICE_REV from device. Verify the values matches the HEX file.

3. **Read and parse one line from HEX file. Write to device.**
   - This step must be repeated for all configuration lines in the HEX file.

4. **Verify programming success.**
   a. Poll PROGRAMMER_STATUS register to confirm programming is complete.
   b. Read the BANK_STATUS register to confirm all configurations were programmed successfully.

Step 1a – Write to DMA Address Register

To read the number of available NVM slots, first set the DMA address, as shown below.



DMA Address Register (least-significant byte first)

S	Address	A	Code	A	Data[0]	A	Data[1]	A	P
	0xC0		0xF7		0x80*		0x10*		

PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

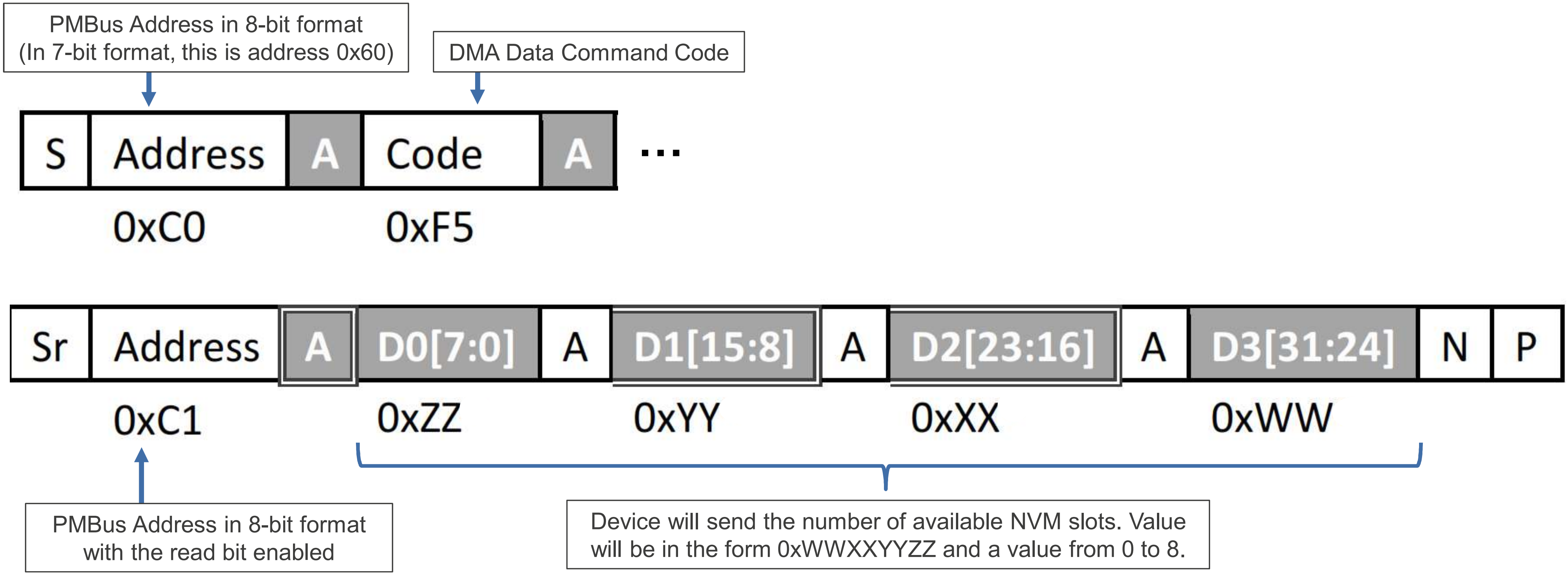
DMA Address Command Code

DMA Address Register Data

DMA Address Register Data

Step 1b – Read DMA Data Register

Next, Read the content of the register pointed to in step 1.



PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

DMA Data Command Code

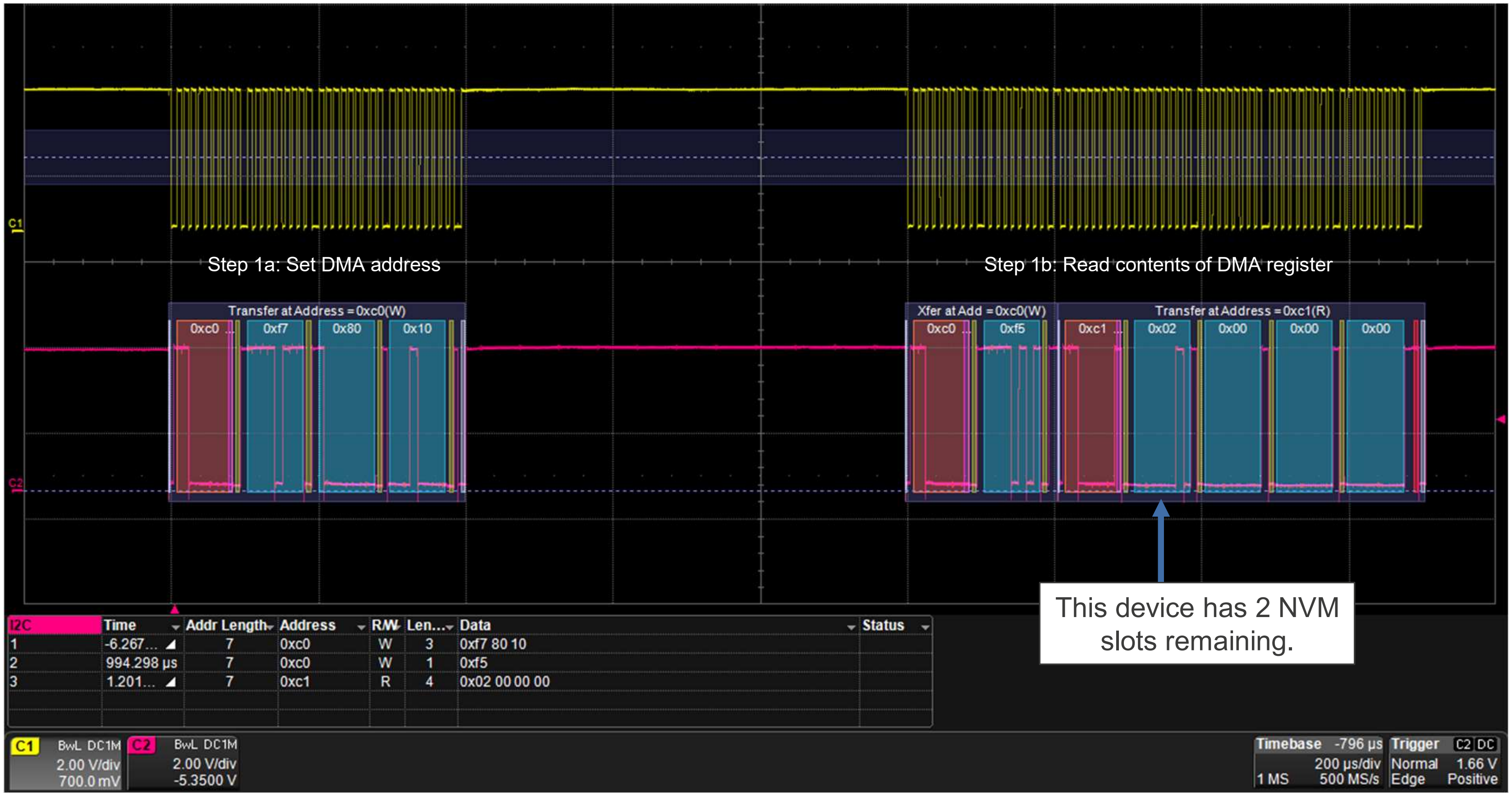
S	Address	A	Code	A	…
	0xC0		0xF5		

Sr	Address	A	D0[7:0]	A	D1[15:8]	A	D2[23:16]	A	D3[31:24]	N	P
	0xC1		0xZZ		0xYY		0xXX		0xWW		

PMBus Address in 8-bit format
with the read bit enabled

Device will send the number of available NVM slots. Value
will be in the form 0xWWXXYYZZ and a value from 0 to 8.

Step 1 – Example Waveforms


C1
C2
Step 1a: Set DMA address
Step 1b: Read contents of DMA register
Transfer at Address = 0xc0(W)
0xc0
0xf7
0x80
0x10
Xfer at Add = 0xc0(W)
0xc0
0xf5
Transfer at Address = 0xc1(R)
0xc1
0x02
0x00
0x00
0x00

I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-6.267...	7	0xc0	W	3	0xf7 80 10	
2	994.298 µs	7	0xc0	W	1	0xf5	
3	1.201...	7	0xc1	R	4	0x02 00 00 00	

C1 BwL DC1M 2.00 V/div 700.0 mV

C2 BwL DC1M 2.00 V/div -5.3500 V

Timebase -796 µs 200 µs/div 1 MS 500 MS/s

Trigger C2 DC Normal 1.66 V Edge Positive

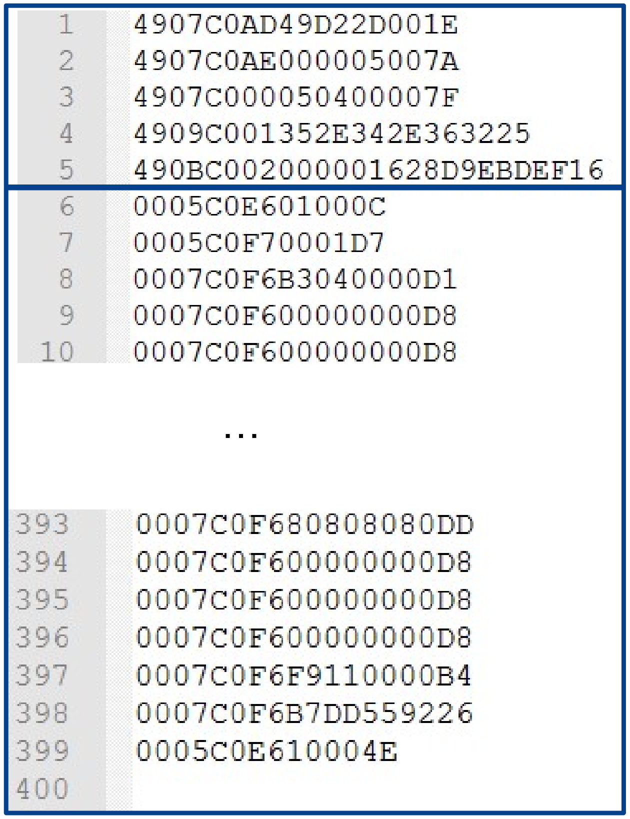
This device has 2 NVM slots remaining.

Step 2a – Parse Header in HEX File

- The first 5 lines in the HEX file (starting with 0x49) are part of the HEX file header and should not be written to the device.

- The HEX header contains IC_DEVICE_ID, IC_DEVICE_REV and HEX_VERSION information.
  - IC_DEVICE_ID in HEX file must match IC_DEVICE_ID read back from device (see step 2b).
  - IC_DEVICE_REV in HEX file must match IC_DEVICE_REV read back from device (see step 2c).
  - HEX_VERSION should be 0x0504000 for all X+Y=4 controllers and 0x07070000 for all X+Y=7 controllers.

Step 2a – Example HEX File



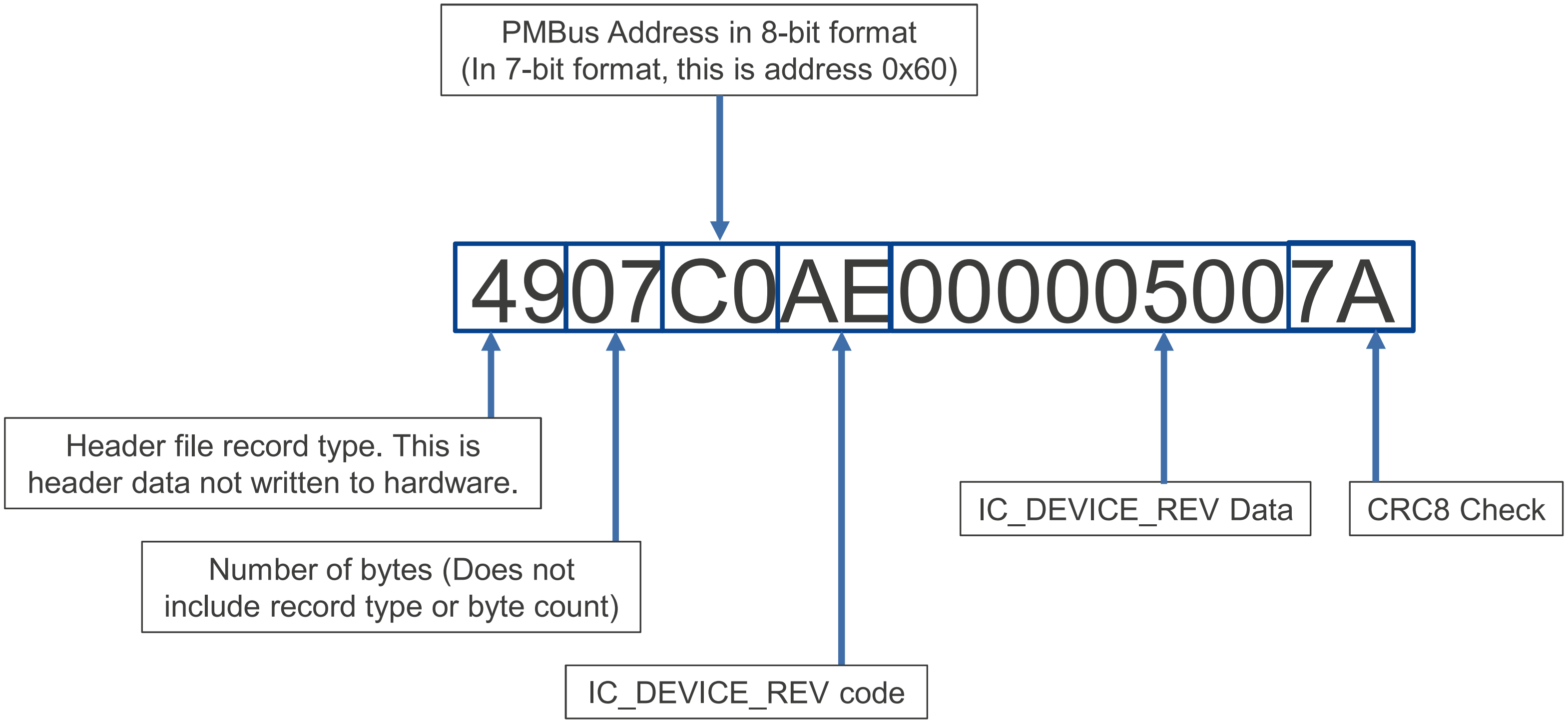
```
1    4907C0AD49D22D001E
2    4907C0AE000005007A
3    4907C000050400007F
4    4909C001352E342E363225
5    490BC002000001628D9EBDEF16
6    0005C0E601000C
7    0005C0F70001D7
8    0007C0F6B3040000D1
9    0007C0F600000000D8
10   0007C0F600000000D8

          ...

393  0007C0F680808080DD
394  0007C0F600000000D8
395  0007C0F600000000D8
396  0007C0F600000000D8
397  0007C0F6F9110000B4
398  0007C0F6B7DD559226
399  0005C0E610004E
400
```

Parse first two lines and verify the HEX file was generated for the device being programmed (see steps 2b and 2c).

- Line 1 = IC_DEVICE_ID
- Line 2 = IC_DEVICE_REV
- Line 3 = HEX_VERSION

Value contained in file	Record Type	Command Code
IC_DEVICE_ID	0x49	0xAD
IC_DEVICE_REV	0x49	0xAE
HEX_VERSION	0x49	0x00

Step 2a – Example HEX File Header Decode



4907C0AE000005007A

PMBus Address in 8-bit format (In 7-bit format, this is address 0x60)

Header file record type. This is header data not written to hardware.

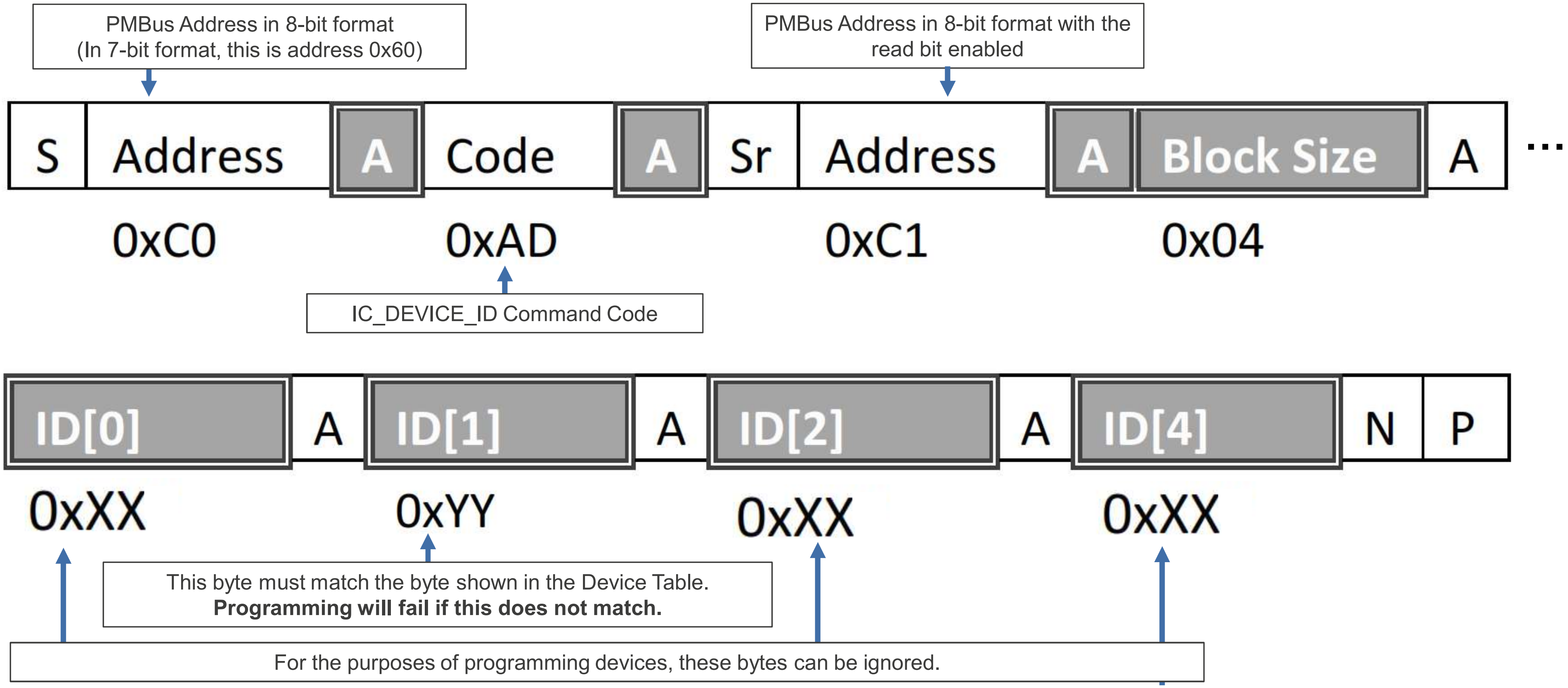
Number of bytes (Does not include record type or byte count)

IC_DEVICE_REV code

IC_DEVICE_REV Data

CRC8 Check

Step 2b – Read IC_DEVICE_ID from Device



PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

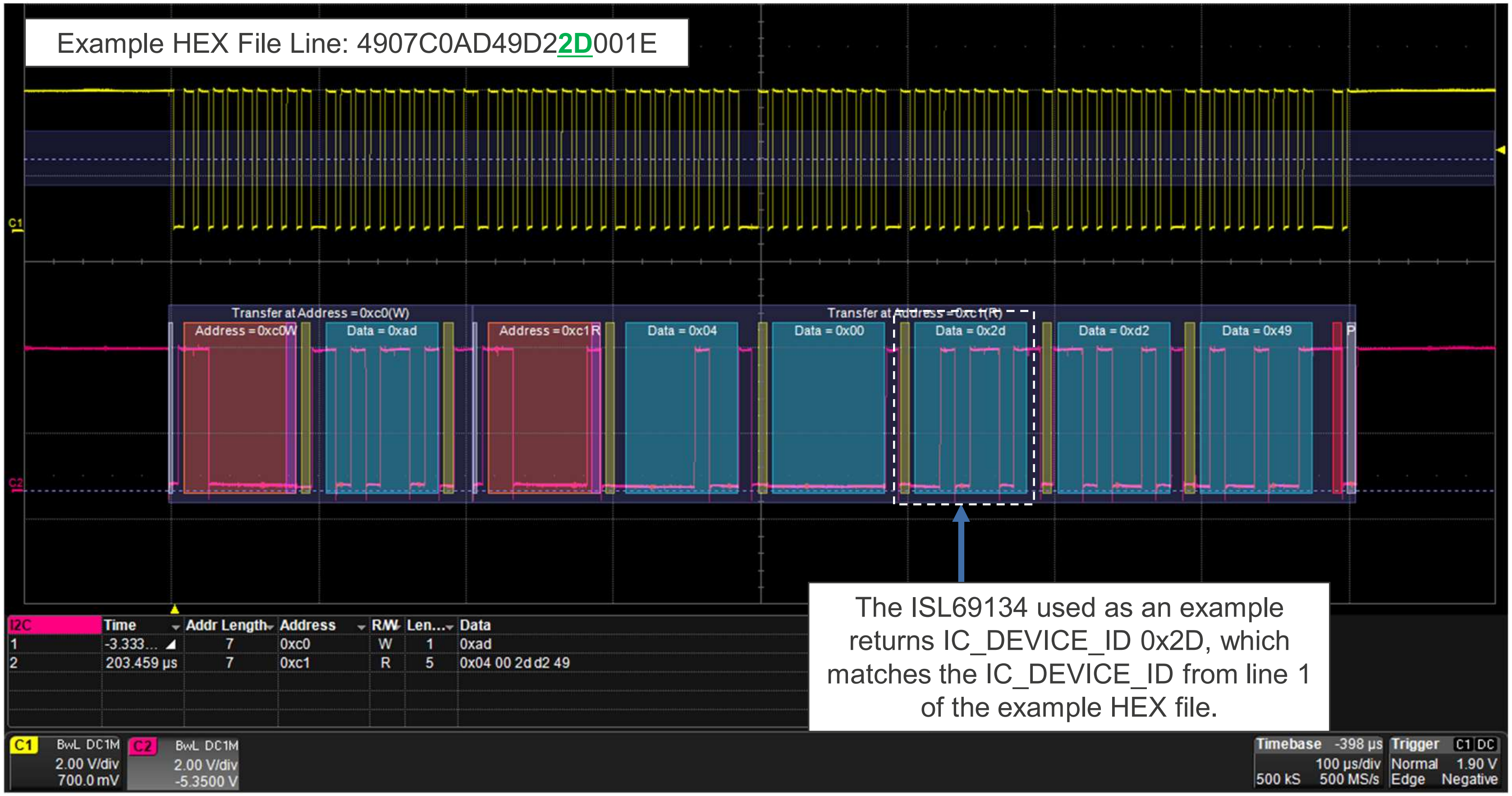
PMBus Address in 8-bit format with the read bit enabled

IC_DEVICE_ID Command Code

This byte must match the byte shown in the Device Table.
Programming will fail if this does not match.

For the purposes of programming devices, these bytes can be ignored.

Step 2b – IC_DEVICE_ID Example Waveform



Example HEX File Line: 4907C0AD49D2**2D**001E

The ISL69134 used as an example returns IC_DEVICE_ID 0x2D, which matches the IC_DEVICE_ID from line 1 of the example HEX file.

I2C	Time	Addr Length	Address	R/W	Len...	Data
1	-3.333...	7	0xc0	W	1	0xad
2	203.459 µs	7	0xc1	R	5	0x04 00 2d d2 49

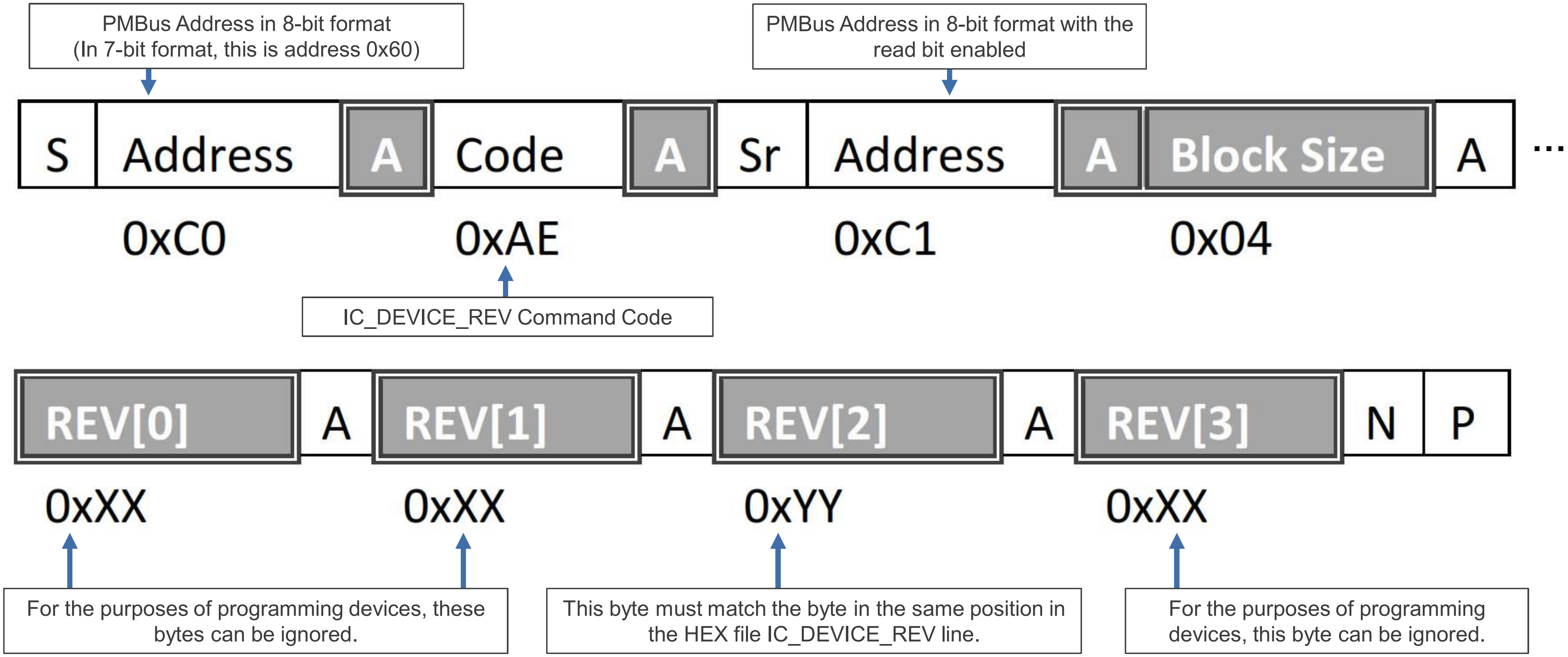
C1 BwL DC1M 2.00 V/div 700.0 mV

C2 BwL DC1M 2.00 V/div -5.3500 V

Timebase -398 µs 100 µs/div 500 kS 500 MS/s

Trigger C1 DC Normal 1.90 V Edge Negative

Step 2c – Read IC_DEVICE_REV from Device



PMBus Address in 8-bit format (In 7-bit format, this is address 0x60)

PMBus Address in 8-bit format with the read bit enabled

S	Address	A	Code	A	Sr	Address	A	Block Size	A	…
	0xC0		0xAE			0xC1		0x04		

IC_DEVICE_REV Command Code

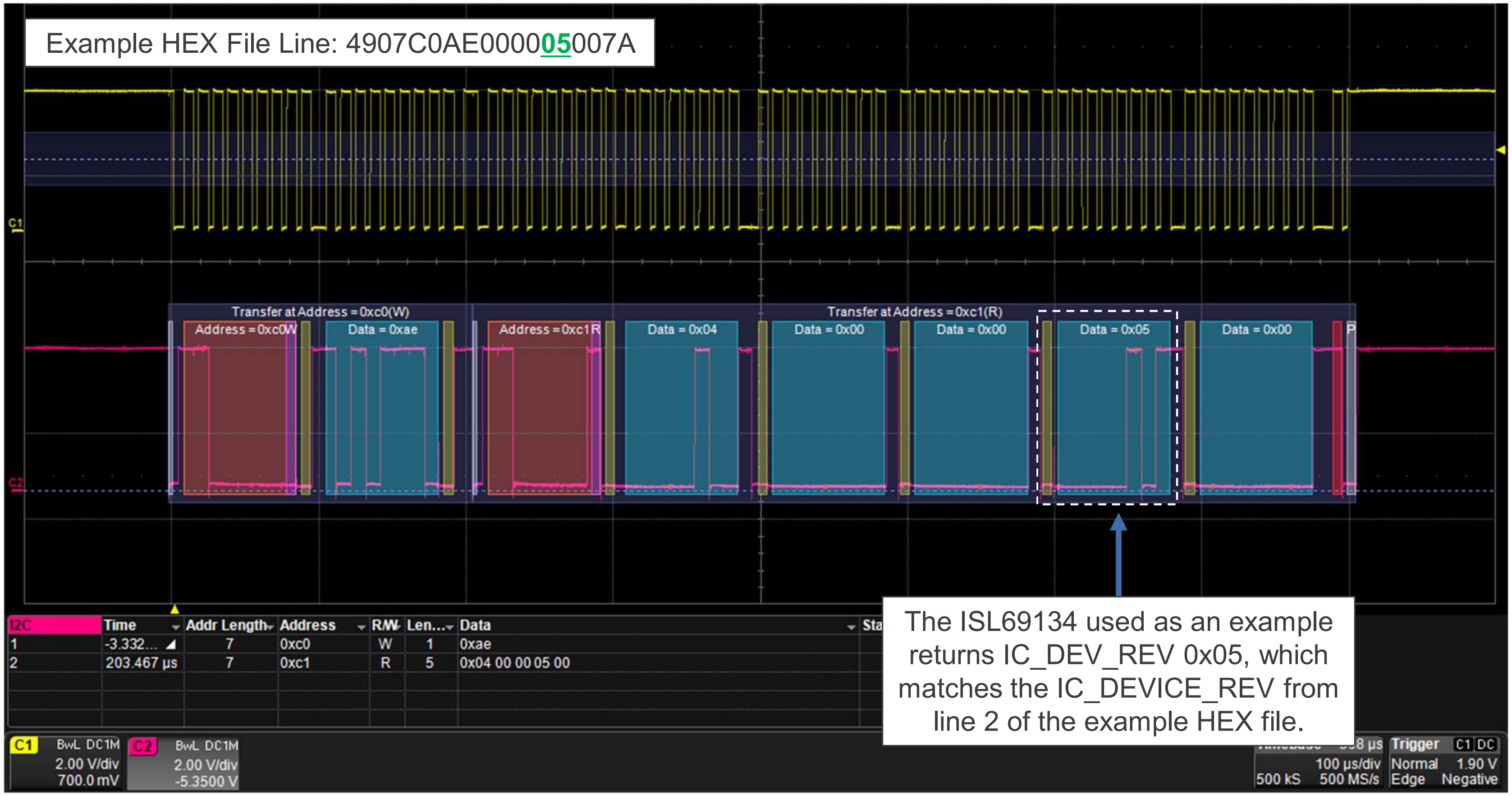
REV[0]	A	REV[1]	A	REV[2]	A	REV[3]	N	P
0xXX		0xXX		0xYY		0xXX		

For the purposes of programming devices, these bytes can be ignored.

This byte must match the byte in the same position in the HEX file IC_DEVICE_REV line.

For the purposes of programming devices, this byte can be ignored.

Step 2c – Example Waveforms



Example HEX File Line: 4907C0AE0000**05**007A

I2C	Time	Addr Length	Address	R/W	Len...	Data	Sta
1	-3.332...	7	0xc0	W	1	0xae	
2	203.467 µs	7	0xc1	R	5	0x04 00 00 05 00	

The ISL69134 used as an example returns IC_DEV_REV 0x05, which matches the IC_DEVICE_REV from line 2 of the example HEX file.

Step 2d – Compare data to HEX file

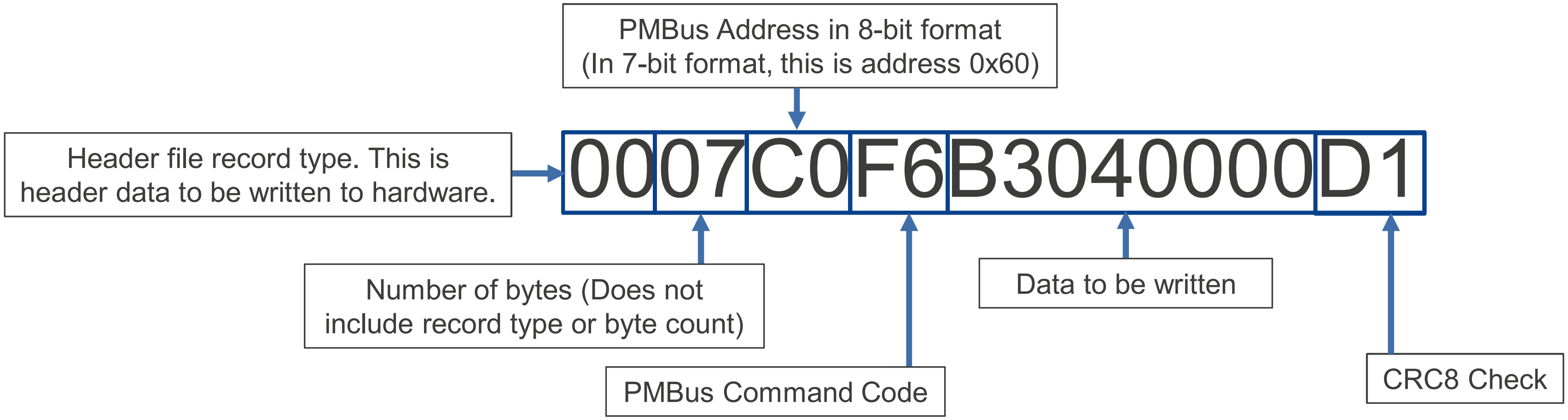
- If the IC_DEVICE_DE, IC_DEVIC_REV bytes in the HEX file match the bytes ready back from the device, proceed to step 3.

- If they do not match, the HEX file was generated for a different device and device programming should be halted.

Step 3 – Parse HEX File and Write to Hardware

Parse remaining lines from HEX file (all lines not starting with 0x49), and write to device

Example HEX File Line:



PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

Header file record type. This is header data to be written to hardware.

00 07 C0 F6 B3040000 D1

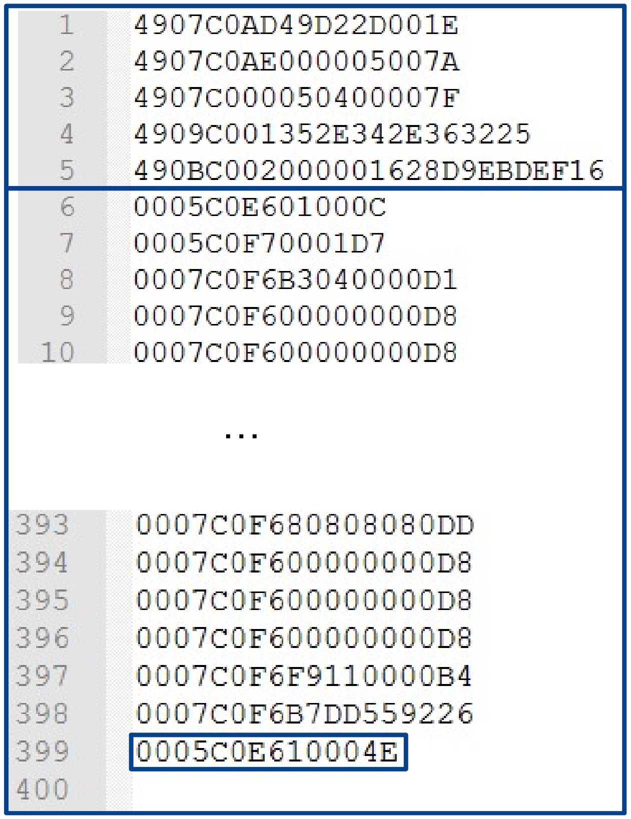
Number of bytes (Does not include record type or byte count)

PMBus Command Code

Data to be written

CRC8 Check

Step 3 – Example HEX File



```
4907C0AD49D22D001E
4907C0AE000005007A
4907C000050400007F
4909C001352E342E363225
490BC002000001628D9EBDEF16
0005C0E601000C
0005C0F70001D7
0007C0F6B3040000D1
0007C0F600000000D8
0007C0F600000000D8

...

0007C0F680808080DD
0007C0F600000000D8
0007C0F600000000D8
0007C0F600000000D8
0007C0F6F9110000B4
0007C0F6B7DD559226
0005C0E610004E
```

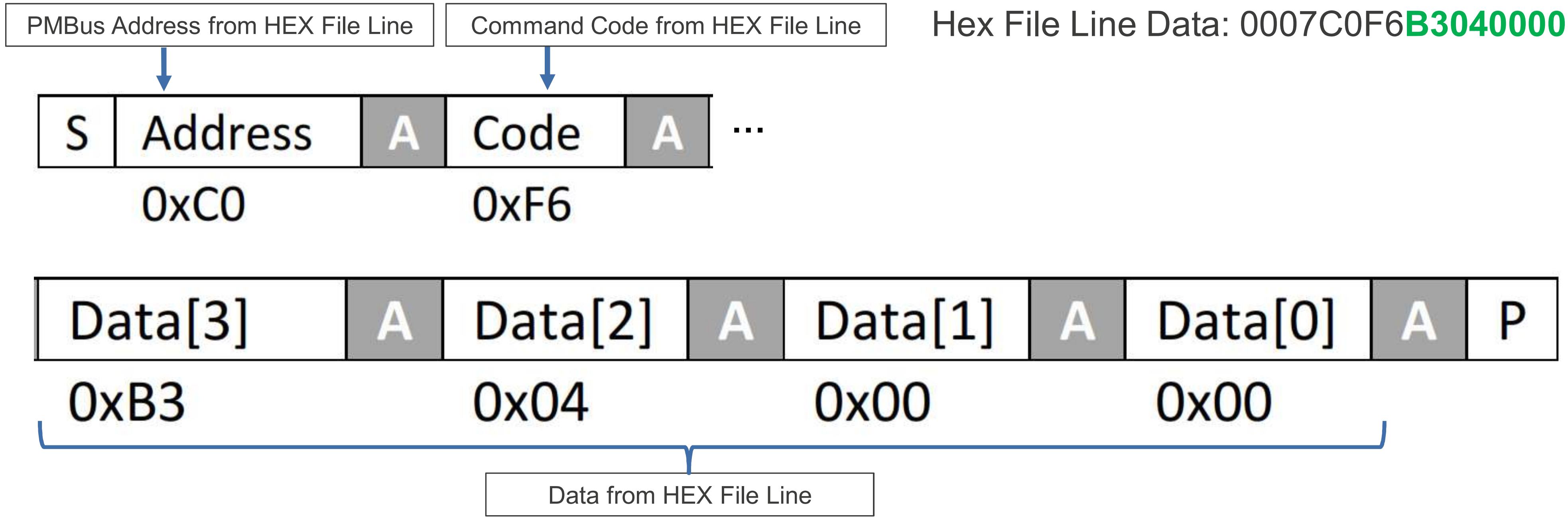
Step 3: Parse and write to device all lines in HEX file that do not start with 0x49

When last line has been written to device, Step 3 is complete.

Step 3 – Example Write of HEX File Data Line

Example PMBus Command:

Hex File Line Data: 0007C0F6B3040000D1



PMBus Address from HEX File Line

Command Code from HEX File Line

S | Address | A | Code | A | …

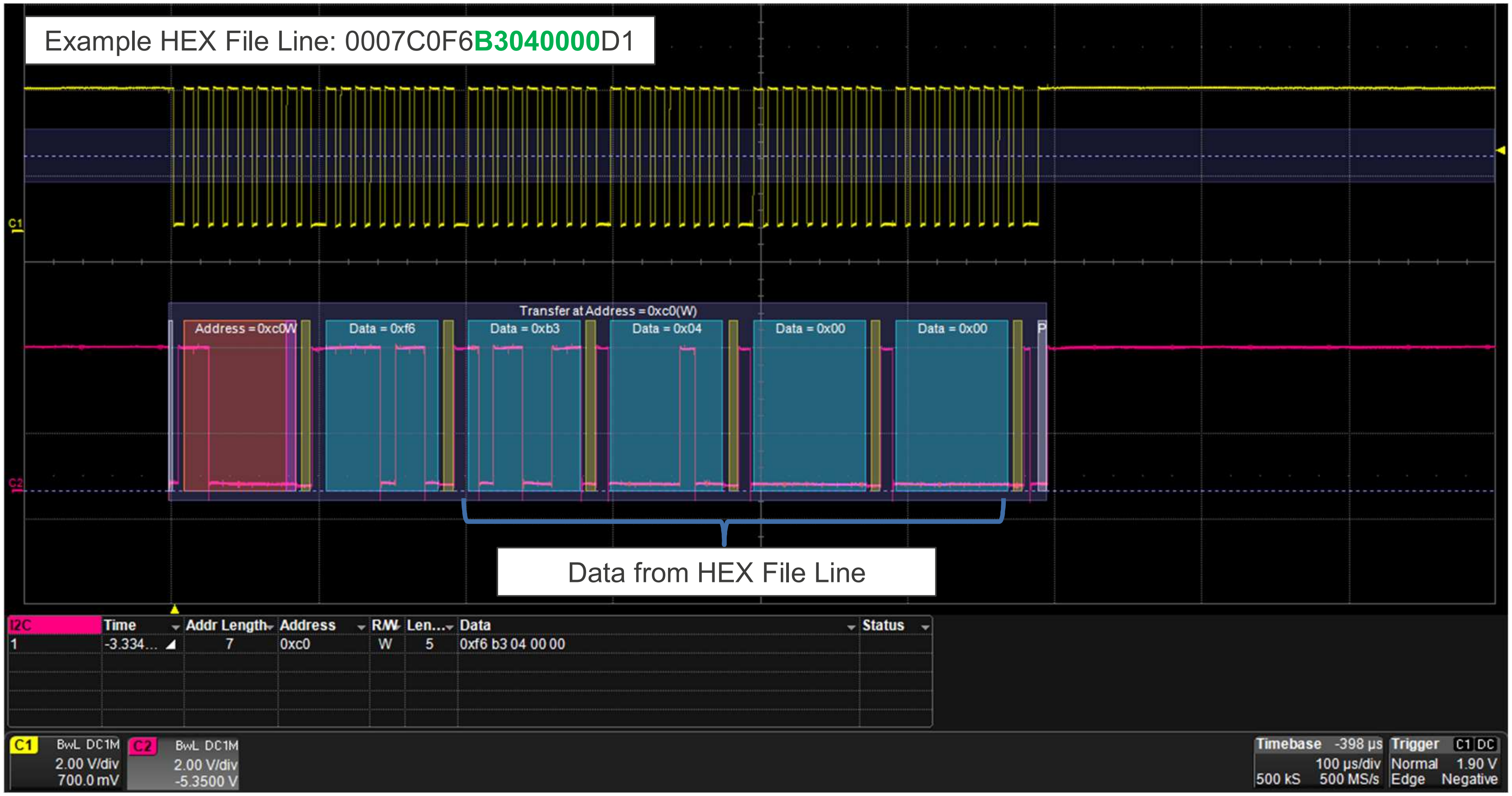
0xC0 0xF6

Data[3] | A | Data[2] | A | Data[1] | A | Data[0] | A | P

0xB3 0x04 0x00 0x00

Data from HEX File Line

Step 3 – Example Write of Hex File Data Line



Example HEX File Line: 0007C0F6**B3040000**D1

Data from HEX File Line

I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-3.334...	7	0xc0	W	5	0xf6 b3 04 00 00	

C1 BwL DC1M 2.00 V/div 700.0 mV

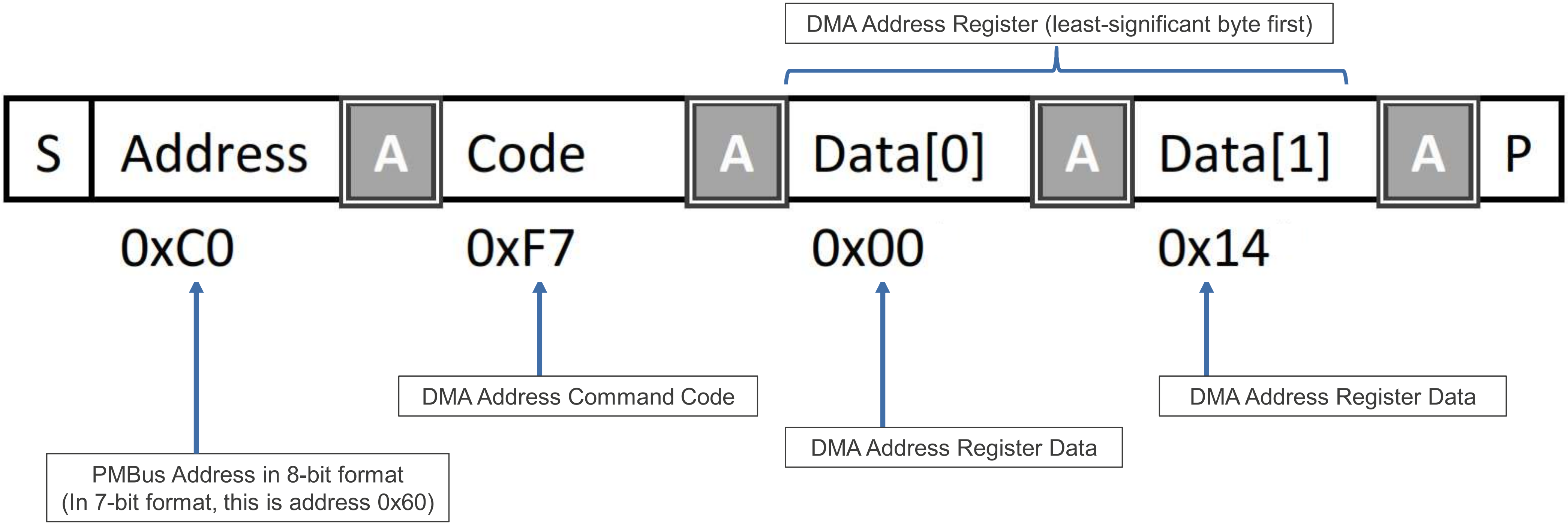
C2 BwL DC1M 2.00 V/div -5.3500 V

Timebase -398 µs 100 µs/div 500 kS 500 MS/s

Trigger C1 DC Normal 1.90 V Edge Negative

Step 4a – Poll PROGRAMMER_STATUS Register

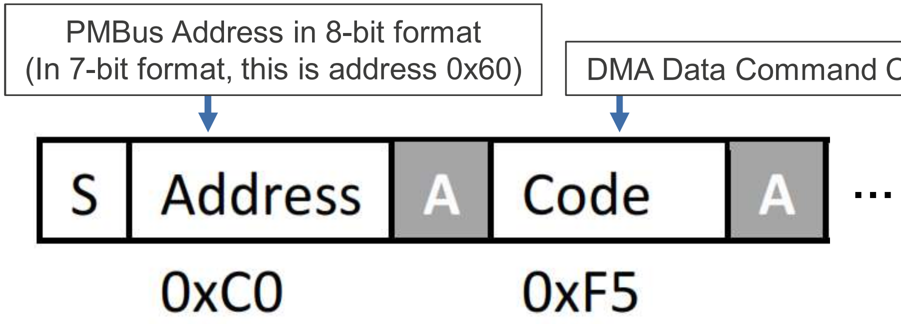
To check for the completion of device programming, first write to the DMA address register as shown below then read the DMA data register until bit 0 is set to 1.


DMA Address Register (least-significant byte first)
S
Address
A
Code
A
Data[0]
A
Data[1]
A
P
0xC0
0xF7
0x00
0x14
DMA Address Command Code
DMA Address Register Data
DMA Address Register Data
PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

Step 4b – Poll PROGRAMMER_STATUS Register

After completing step 4a, use the DMA read command to poll register until bit 0 is set to 1.

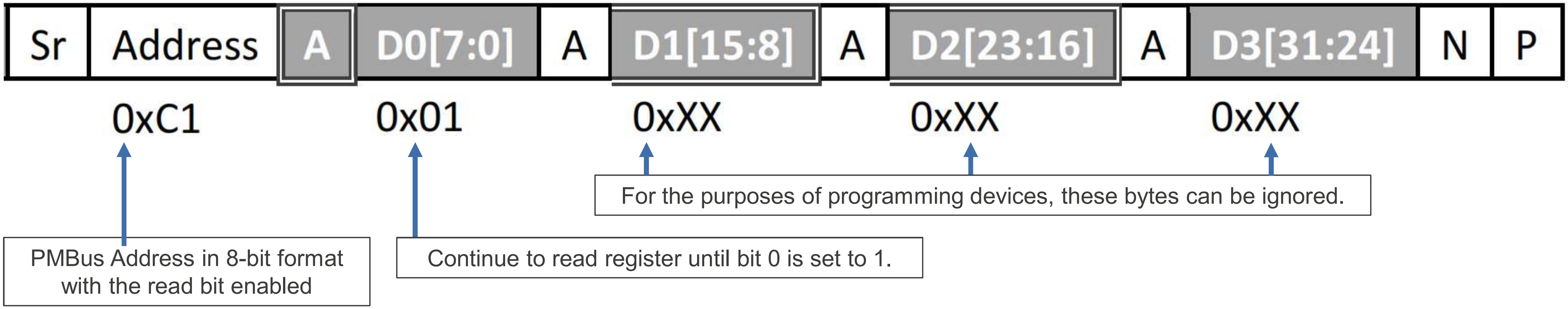
If after 2s timeout, bit 0 has not been set to 1, the part has failed programming. See step 4c for more details.



PMBus Address in 8-bit format (In 7-bit format, this is address 0x60)

DMA Data Command Code

S	Address	A	Code	A	…
	0xC0		0xF5		



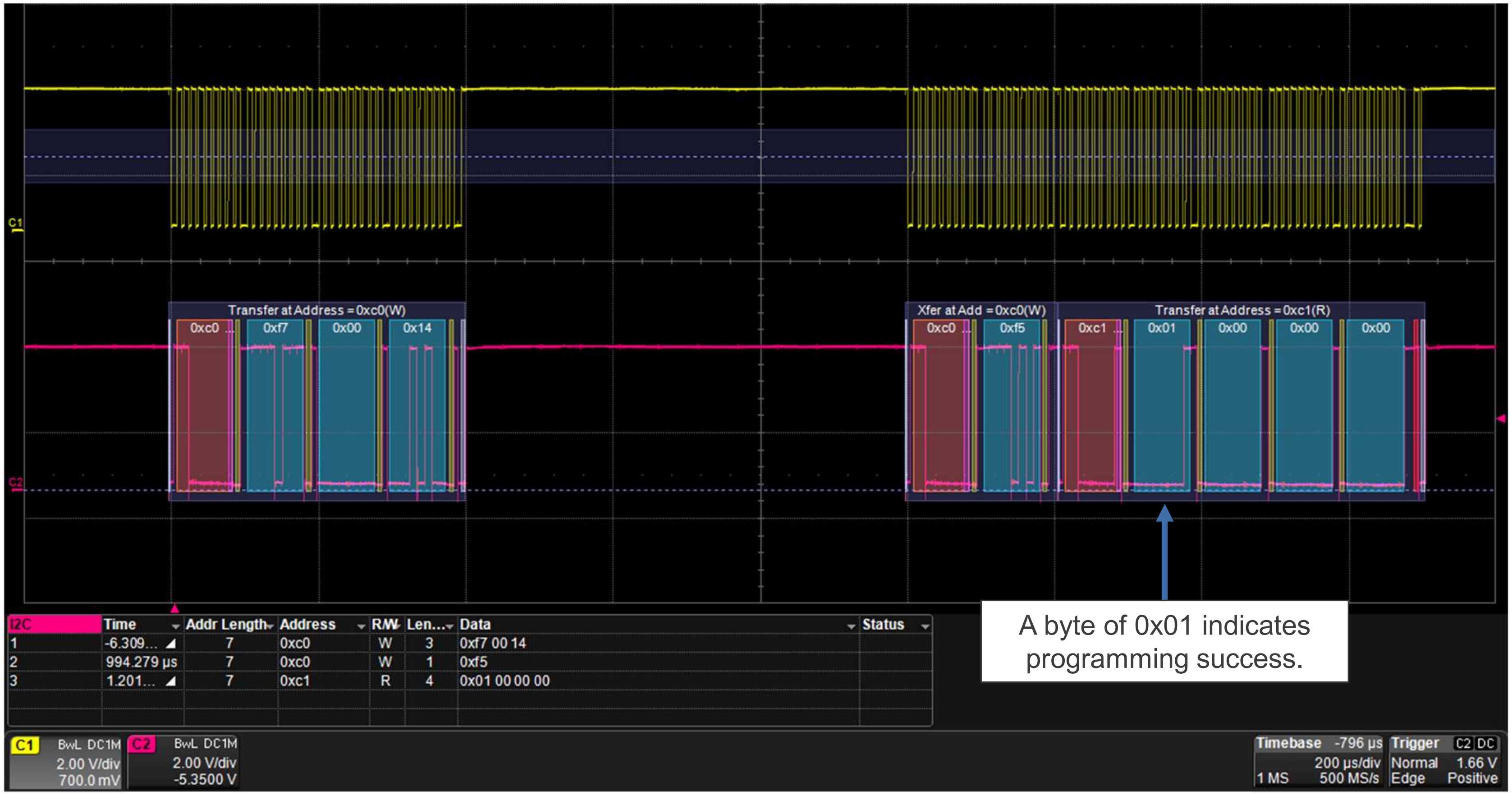
Sr	Address	A	D0[7:0]	A	D1[15:8]	A	D2[23:16]	A	D3[31:24]	N	P
	0xC1		0x01		0xXX		0xXX		0xXX		

For the purposes of programming devices, these bytes can be ignored.

PMBus Address in 8-bit format with the read bit enabled

Continue to read register until bit 0 is set to 1.

Step 4a, 4b – Example Waveforms



I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-6.309...	7	0xc0	W	3	0xf7 00 14	
2	994.279 µs	7	0xc0	W	1	0xf5	
3	1.201...	7	0xc1	R	4	0x01 00 00 00	

C1 BwL DC1M 2.00 V/div 700.0 mV

C2 BwL DC1M 2.00 V/div -5.3500 V

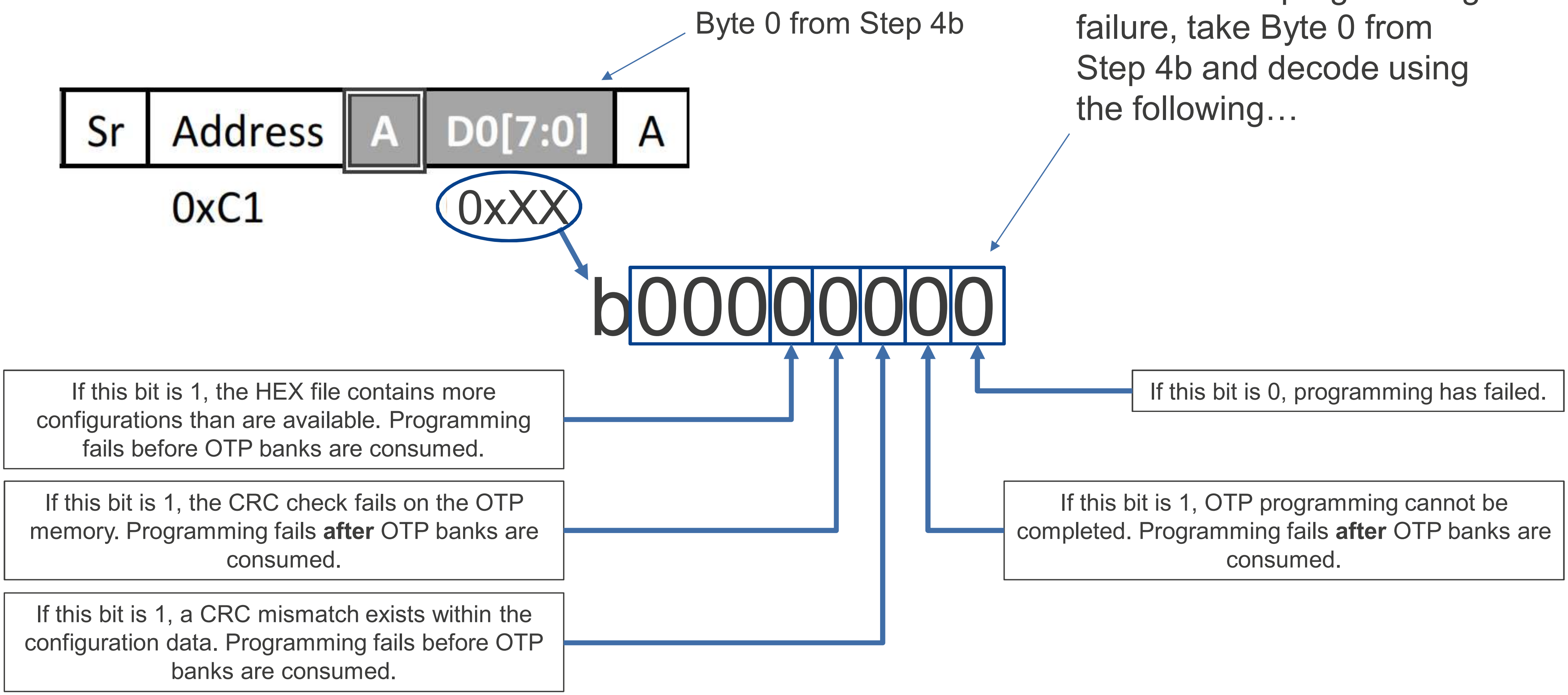
Timebase -796 µs 200 µs/div 1 MS 500 MS/s

Trigger C2 DC Normal 1.66 V Edge Positive

A byte of 0x01 indicates programming success.

Step 4c – Programming Failure

To determine programming failure, take Byte 0 from Step 4b and decode using the following…



If this bit is 1, the HEX file contains more configurations than are available. Programming fails before OTP banks are consumed.

If this bit is 1, the CRC check fails on the OTP memory. Programming fails **after** OTP banks are consumed.

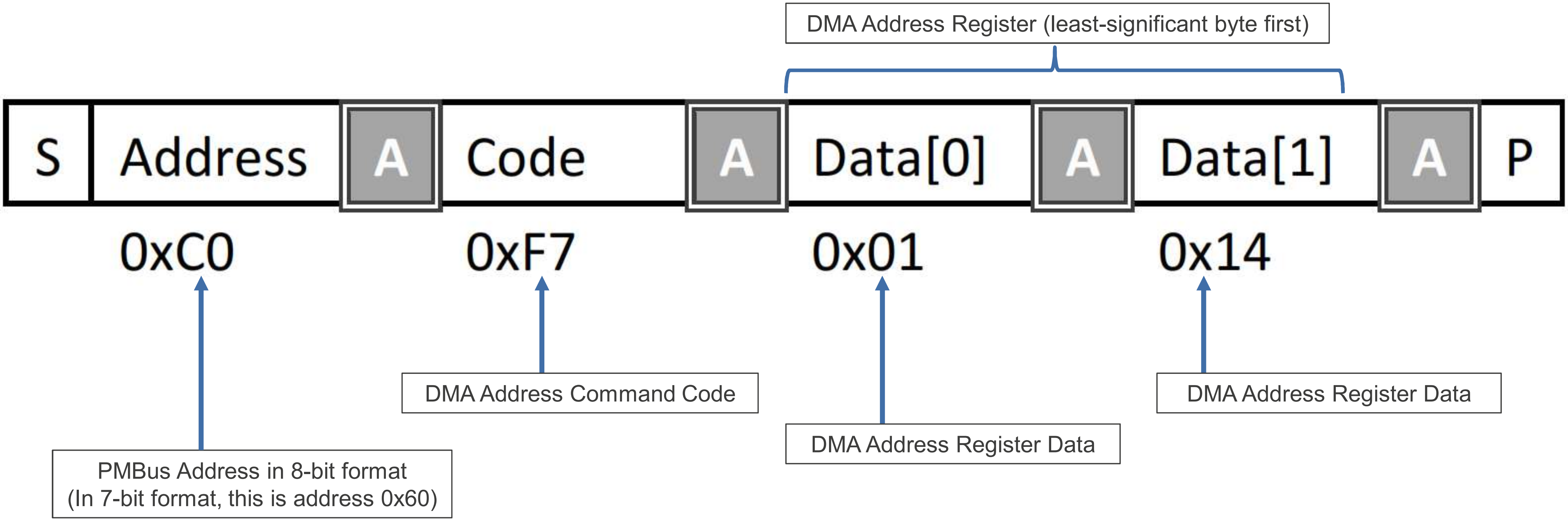
If this bit is 1, a CRC mismatch exists within the configuration data. Programming fails before OTP banks are consumed.

If this bit is 0, programming has failed.

If this bit is 1, OTP programming cannot be completed. Programming fails **after** OTP banks are consumed.

Step 4c – Read BANK_STATUS Register

To read the BANK_STATUS register, first write to the DMA address register as shown below then read the DMA data register to retrieve BANK_STATUS data.



DMA Address Register (least-significant byte first)

S	Address	A	Code	A	Data[0]	A	Data[1]	A	P
	0xC0		0xF7		0x01		0x14		

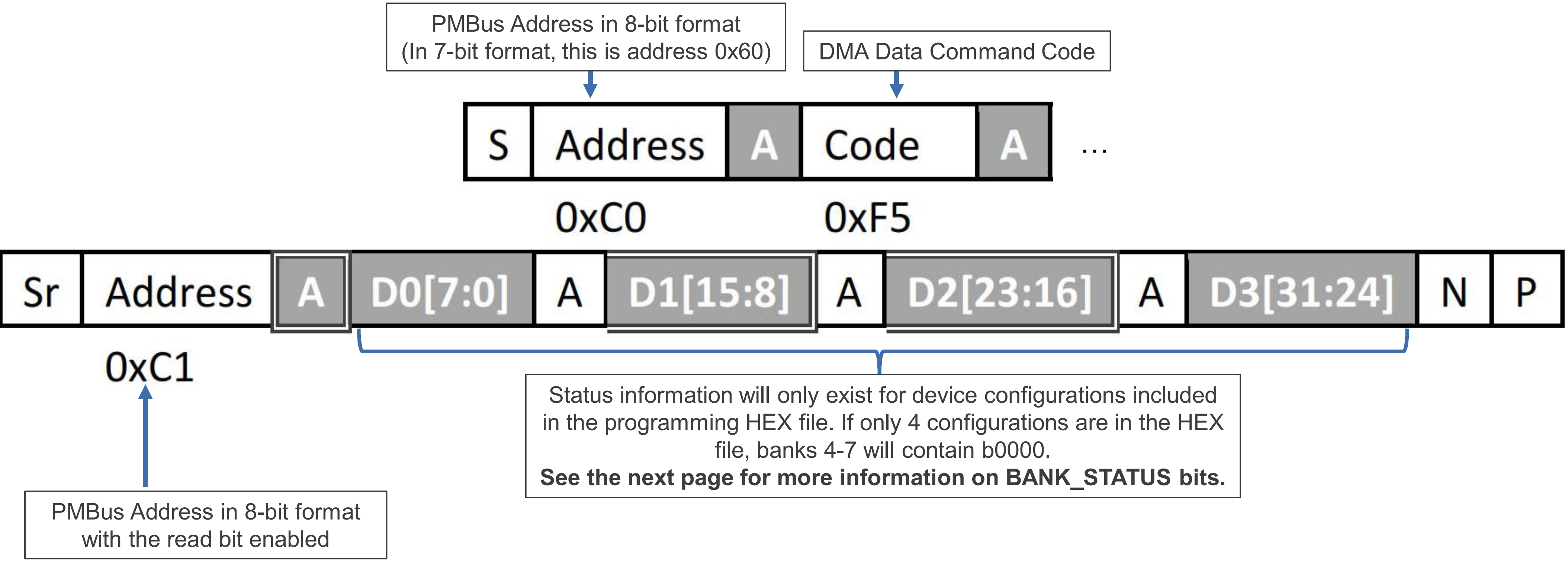
DMA Address Command Code

DMA Address Register Data

DMA Address Register Data

PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

Step 4b – Read BANK_STATUS Register



PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

DMA Data Command Code

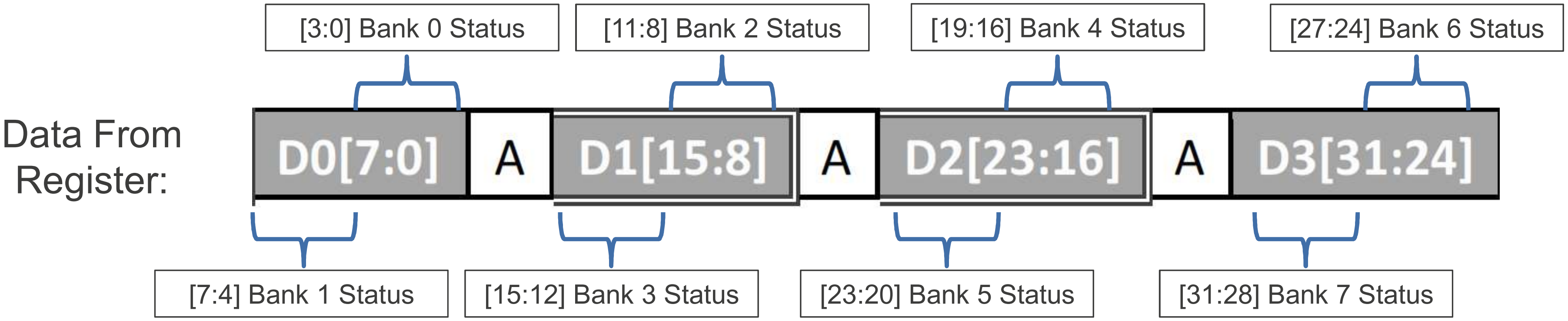
Status information will only exist for device configurations included in the programming HEX file. If only 4 configurations are in the HEX file, banks 4-7 will contain b0000.
See the next page for more information on BANK_STATUS bits.

PMBus Address in 8-bit format
with the read bit enabled

Step 4b – Read BANK_STATUS Register

Every set of bank status bits will correspond to the table below:

Bank Status Bits	Description
b1000	Fail: CRC mismatch OTP
b0100	Fail: CRC mismatch RAM
b0010	Fail: Write error
b0001	Bank Written
b0000	Bank Unaffected



[3:0] Bank 0 Status

[11:8] Bank 2 Status

[19:16] Bank 4 Status

[27:24] Bank 6 Status

Data From Register:

D0[7:0] A D1[15:8] A D2[23:16] A D3[31:24]

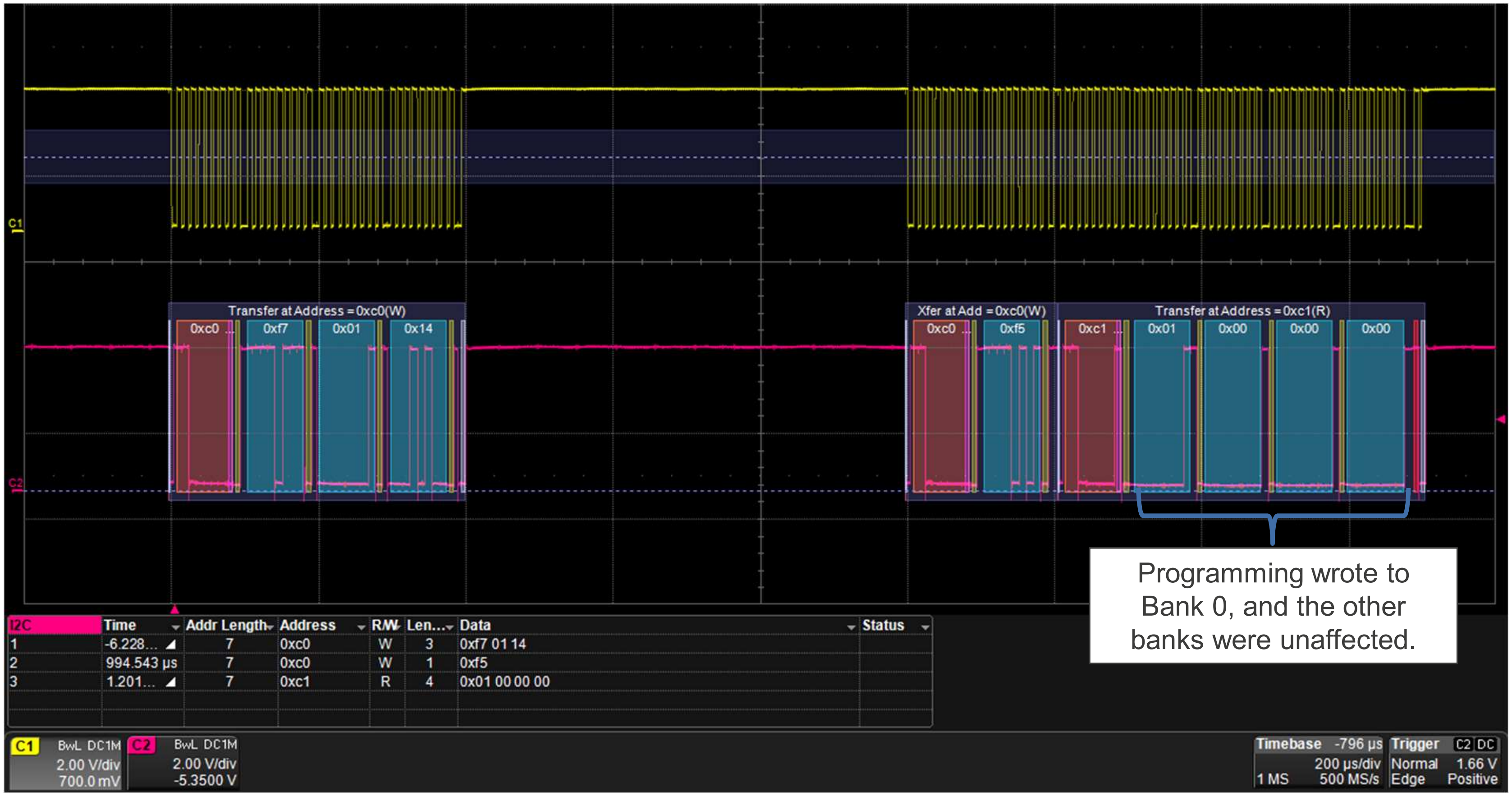
[7:4] Bank 1 Status

[15:12] Bank 3 Status

[23:20] Bank 5 Status

[31:28] Bank 7 Status

Step 4b – Example Waveforms



Programming wrote to Bank 0, and the other banks were unaffected.

I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-6.228...	7	0xc0	W	3	0xf7 01 14	
2	994.543 µs	7	0xc0	W	1	0xf5	
3	1.201...	7	0xc1	R	4	0x01 00 00 00	

C1 BwL DC1M 2.00 V/div 700.0 mV

C2 BwL DC1M 2.00 V/div -5.3500 V

Timebase -796 µs 200 µs/div 1 MS 500 MS/s

Trigger C2 DC Normal 1.66 V Edge Positive

Algorithm Completion

- After successfully completing programming, 3.3V VCC must be powered down to apply changes.

- Reading the new CRC from RAM is possible after cycling VCC with a 50ms delay.

SECTION 2:

REPROGRAMMING DEVICES

ALGORITHM OVERVIEW

1. **Determine number of NVM slots available. (Optional)**

2. **Verify device and file versions.**
   a. Read and parse header data from HEX file. Go to the Step 2a section for more information.
   b. Read IC_DEVICE_ID from device. Verify the value matches the Device Table.
   c. Read IC_DEVICE_REV from device. Verify the values matches the HEX file.

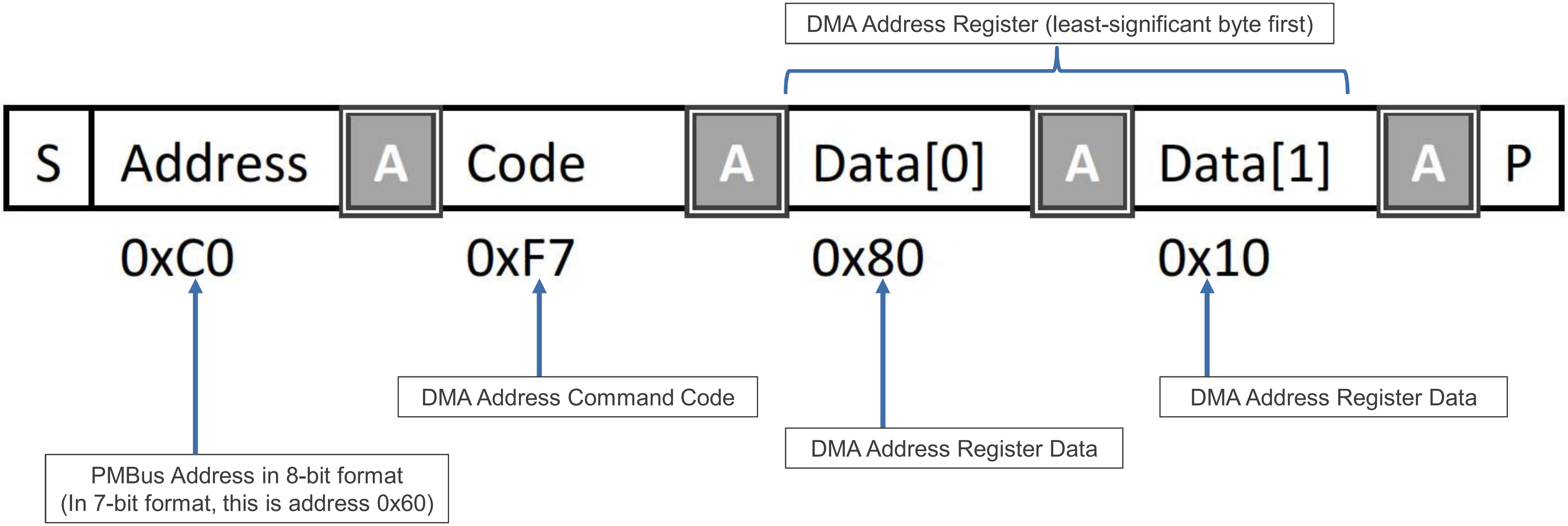
3. **Enable full power mode or voltage regulation.**
   a. If neither rail is enabled, enable full power mode on the part before programming.
   b. If one or both rails are enabled, allow the part to regulate voltage while programming.

4. **Read and parse one line from HEX file. Write to device.**
   - This step must be repeated for all configuration lines in the HEX file.

5. **Verify programming success.**
   a. Poll PROGRAMMER_STATUS register to confirm programming is complete.
   b. Read the BANK_STATUS register to confirm all configurations were programmed successfully.

Step 1 – Write to DMA Address Register

To read the number of available NVM slots, first write to the DMA address register as shown below then read the DMA data register.



DMA Address Register (least-significant byte first)

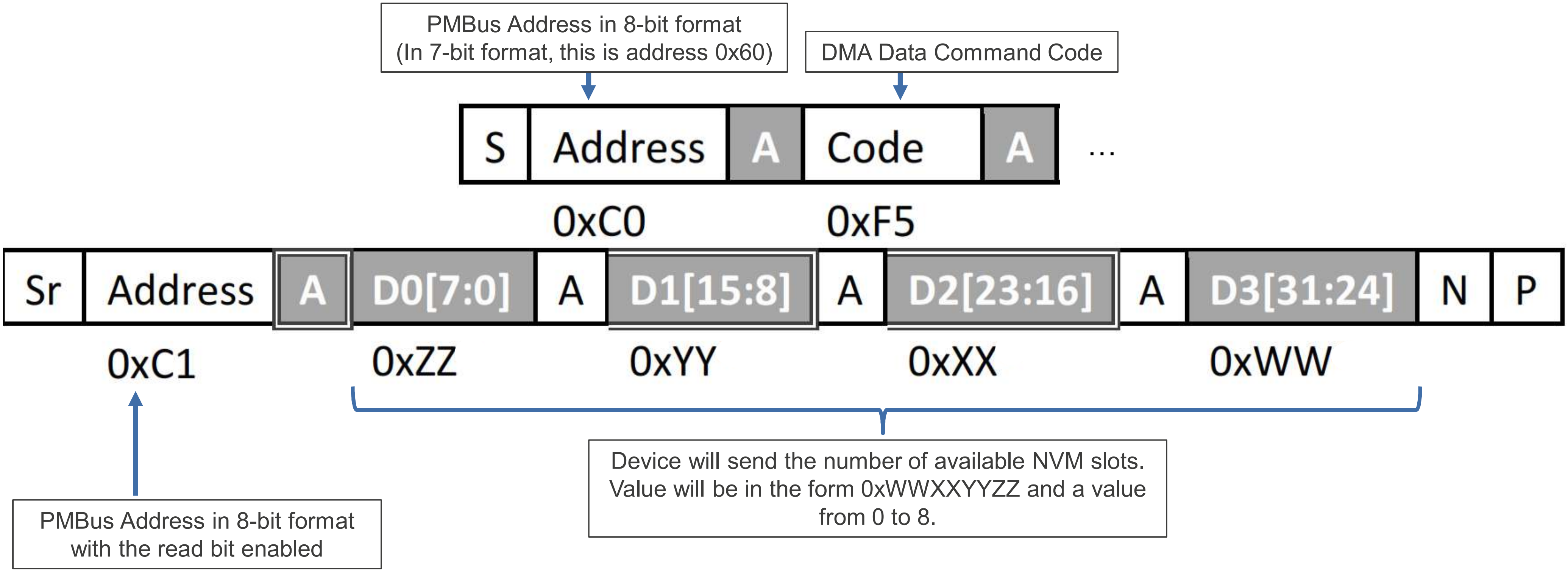
DMA Address Command Code

DMA Address Register Data

DMA Address Register Data

PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

Step 1 – Read DMA Data Register



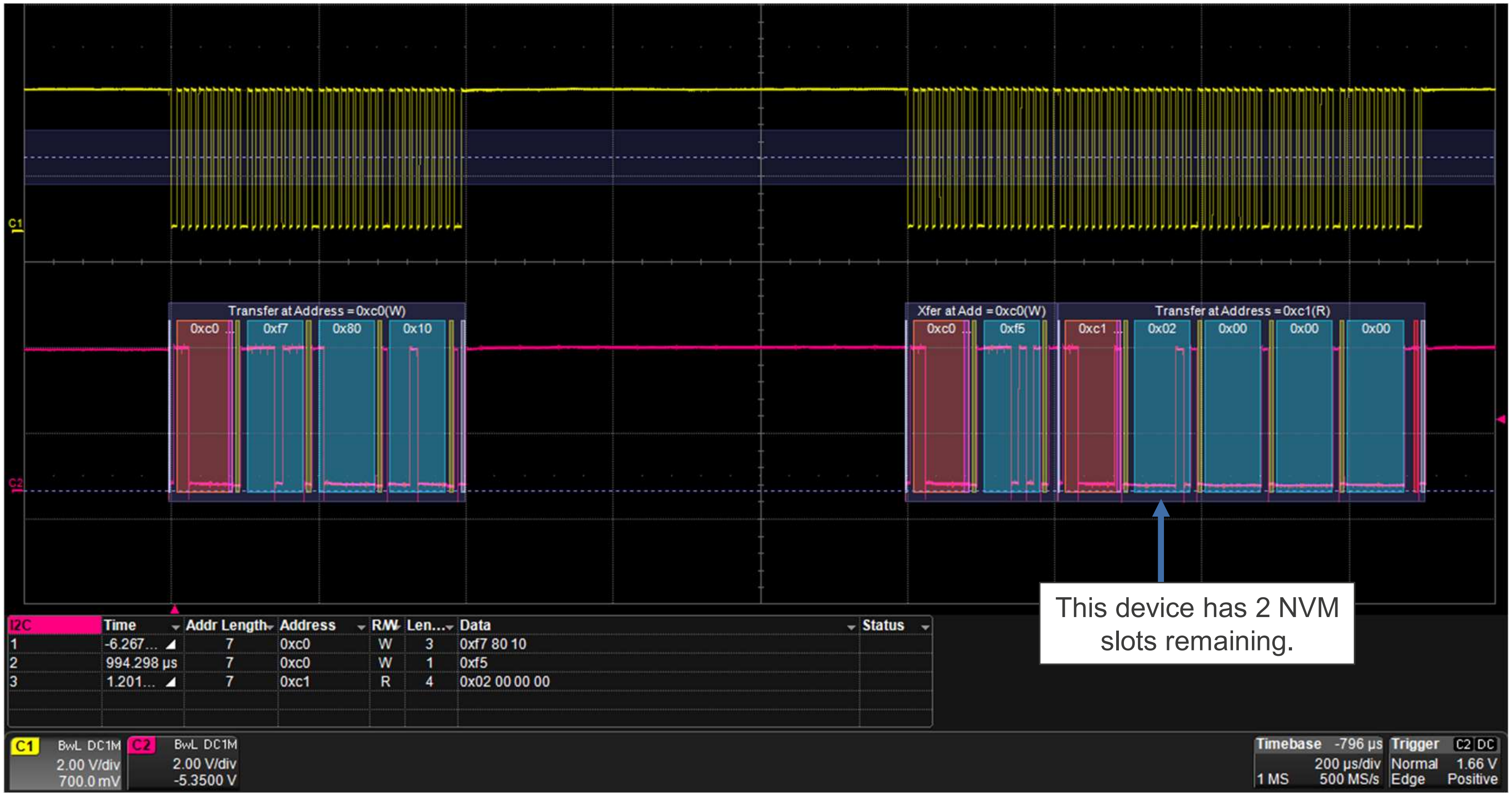
PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

DMA Data Command Code

Device will send the number of available NVM slots. Value will be in the form 0xWWXXYYZZ and a value from 0 to 8.

PMBus Address in 8-bit format
with the read bit enabled

Step 1 – Example Waveforms



I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-6.267...	7	0xc0	W	3	0xf7 80 10	
2	994.298 µs	7	0xc0	W	1	0xf5	
3	1.201...	7	0xc1	R	4	0x02 00 00 00	

C1 BwL DC1M 2.00 V/div 700.0 mV

C2 BwL DC1M 2.00 V/div -5.3500 V

Timebase -796 µs 200 µs/div 1 MS 500 MS/s

Trigger C2 DC Normal 1.66 V Edge Positive

This device has 2 NVM slots remaining.

Step 2a – Parse Header in HEX File

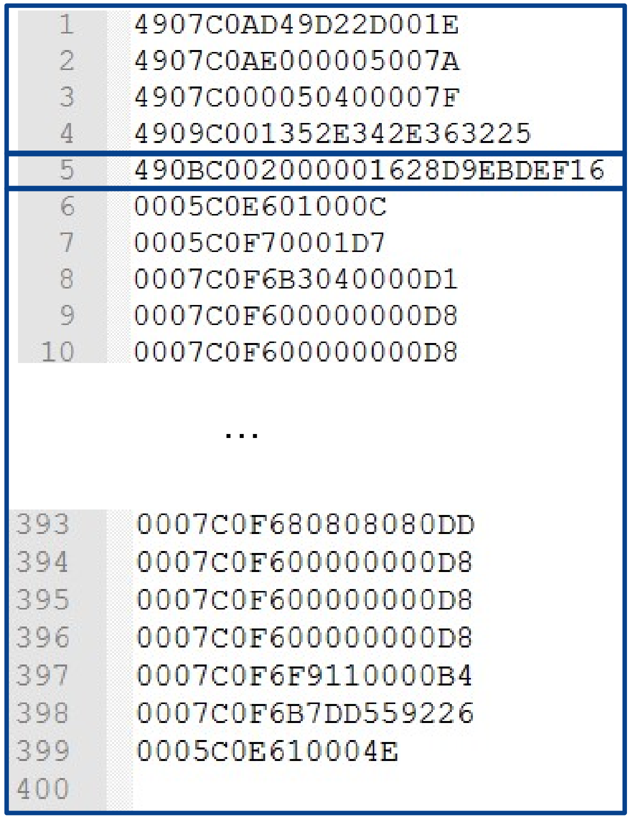
- Any line in the HEX file starting with 0x49 is not written to the device.

- Lines starting with 0x490A or 0x490B can be ignored.

- HEX_VERSION in the file must match the value in the Device Table on page 3 for the device to be programmed.

Value contained in file	Record Type	Command Code
IC_DEVICE_ID	0x49	0xAD
IC_DEVICE_REV	0x49	0xAE
HEX_VERSION	0x49	0x00

Step 2a – Example HEX File



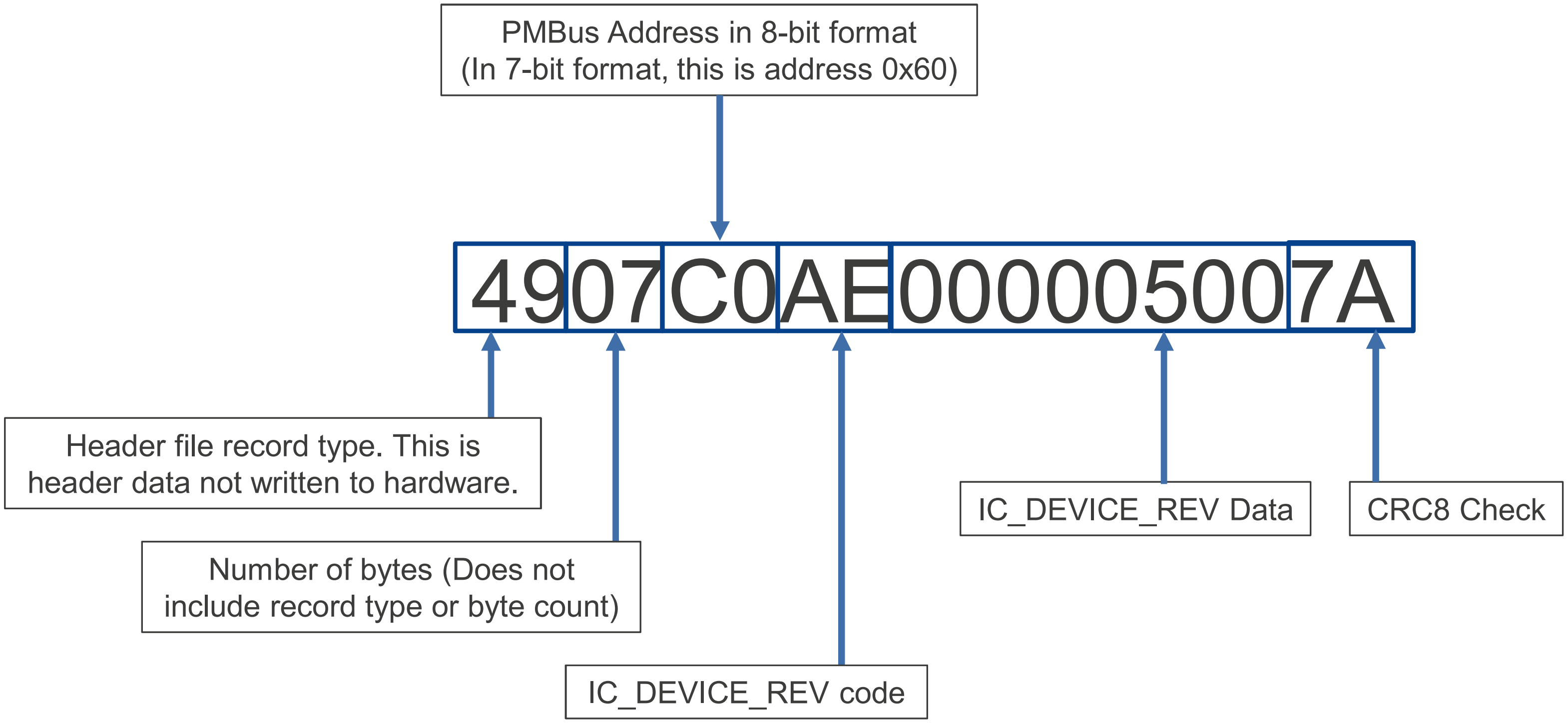
```
4907C0AD49D22D001E
4907C0AE000005007A
4907C000050400007F
4909C001352E342E363225
490BC002000001628D9EBDEF16
0005C0E601000C
0005C0F70001D7
0007C0F6B3040000D1
0007C0F600000000D8
0007C0F600000000D8

...

0007C0F680808080DD
0007C0F600000000D8
0007C0F600000000D8
0007C0F600000000D8
0007C0F6F9110000B4
0007C0F6B7DD559226
0005C0E610004E
```

Parse these four lines.

Ignore lines beginning with 0x490A or 0x490B.

In Step 2a, ignore these lines.

Step 2a – Example HEX File Header



PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

49 07 C0 AE 00000500 7A

Header file record type. This is header data not written to hardware.

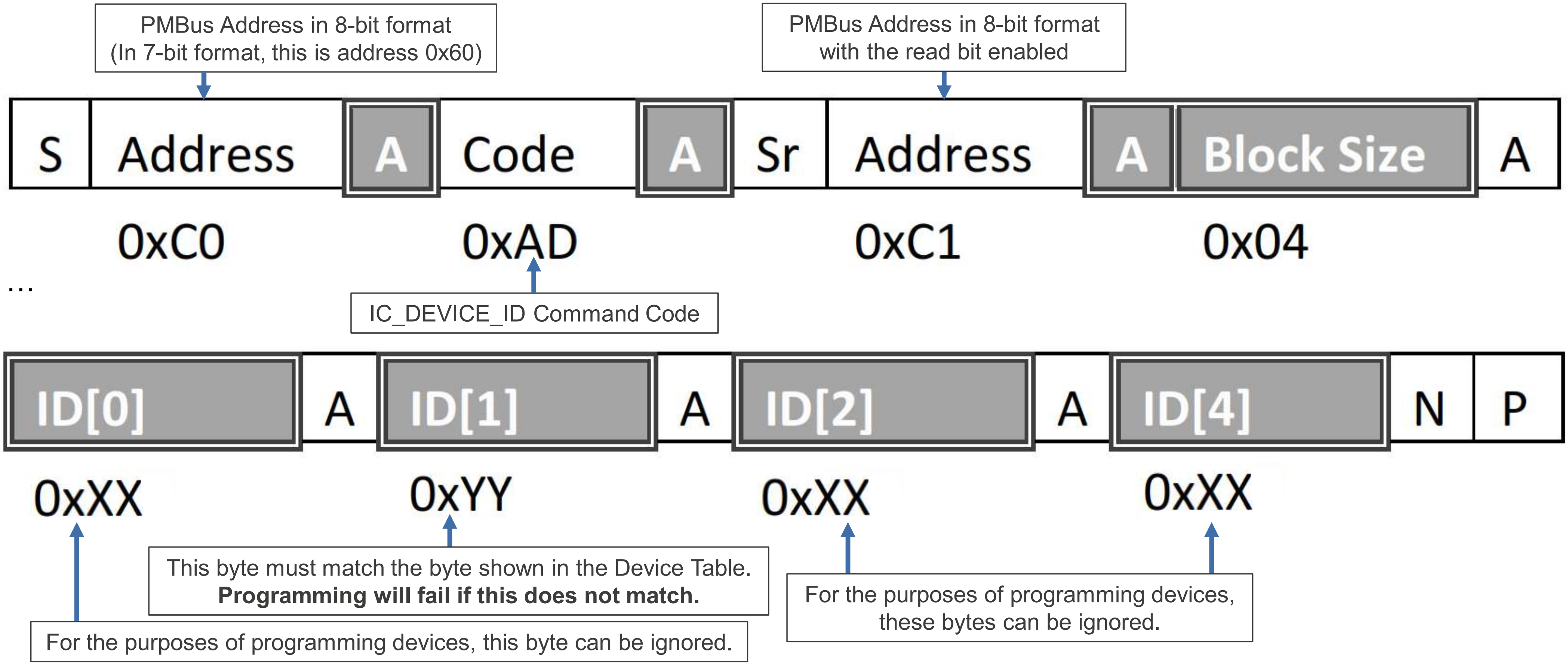
Number of bytes (Does not include record type or byte count)

IC_DEVICE_REV code

IC_DEVICE_REV Data

CRC8 Check

Step 2b – Verify IC_DEVICE_ID



PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

PMBus Address in 8-bit format
with the read bit enabled

S | Address | A | Code | A | Sr | Address | A | Block Size | A

0xC0 0xAD 0xC1 0x04

…

IC_DEVICE_ID Command Code

ID[0] | A | ID[1] | A | ID[2] | A | ID[4] | N | P

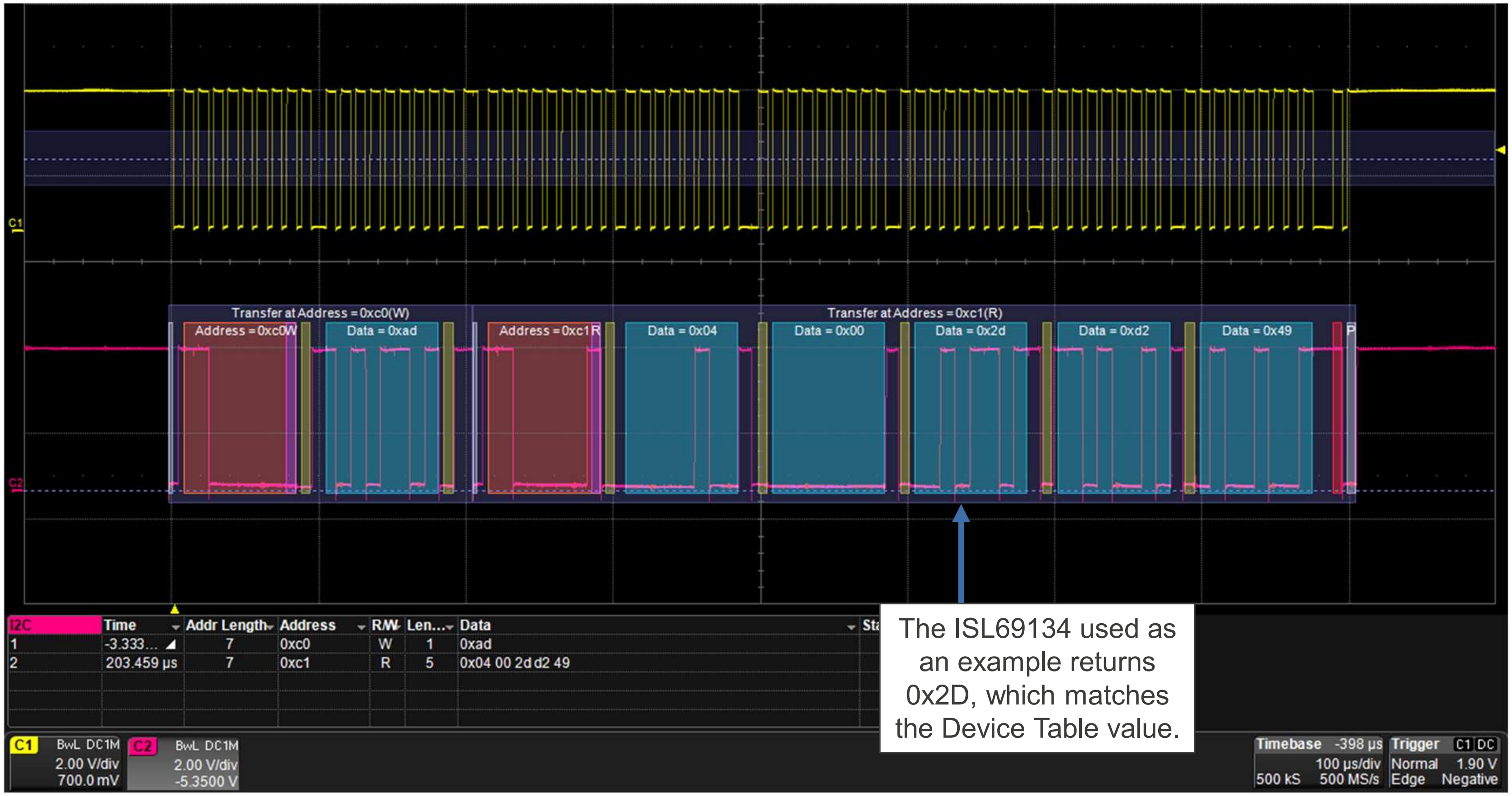
0xXX 0xYY 0xXX 0xXX

This byte must match the byte shown in the Device Table.
Programming will fail if this does not match.

For the purposes of programming devices, this byte can be ignored.

For the purposes of programming devices,
these bytes can be ignored.

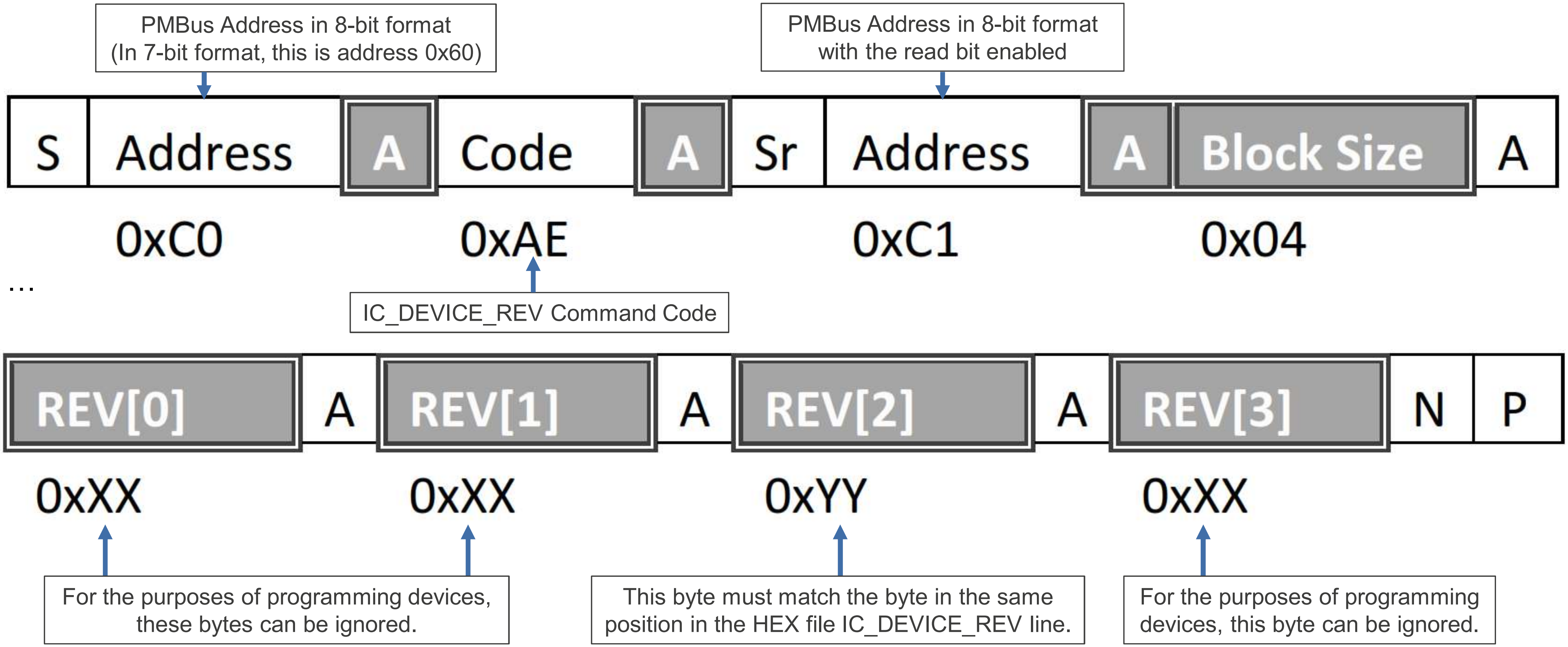
Step 2b – Example Waveforms


C1
C2
Transfer at Address = 0xc0(W)
Address = 0xc0W
Data = 0xad
Address = 0xc1R
Data = 0x04
Transfer at Address = 0xc1(R)
Data = 0x00
Data = 0x2d
Data = 0xd2
Data = 0x49
P
C1 BwL DC1M 2.00 V/div 700.0 mV
C2 BwL DC1M 2.00 V/div -5.3500 V
Timebase -398 µs 100 µs/div 500 kS 500 MS/s
Trigger C1 DC Normal 1.90 V Edge Negative

I2C	Time	Addr Length	Address	R/W	Len...	Data	Sta
1	-3.333...	7	0xc0	W	1	0xad	
2	203.459 µs	7	0xc1	R	5	0x04 00 2d d2 49	

The ISL69134 used as an example returns 0x2D, which matches the Device Table value.

Step 2c – Verify IC_DEVICE_REV



PMBus Address in 8-bit format (In 7-bit format, this is address 0x60)

PMBus Address in 8-bit format with the read bit enabled

S	Address	A	Code	A	Sr	Address	A	Block Size	A
	0xC0		0xAE			0xC1		0x04	

…

IC_DEVICE_REV Command Code

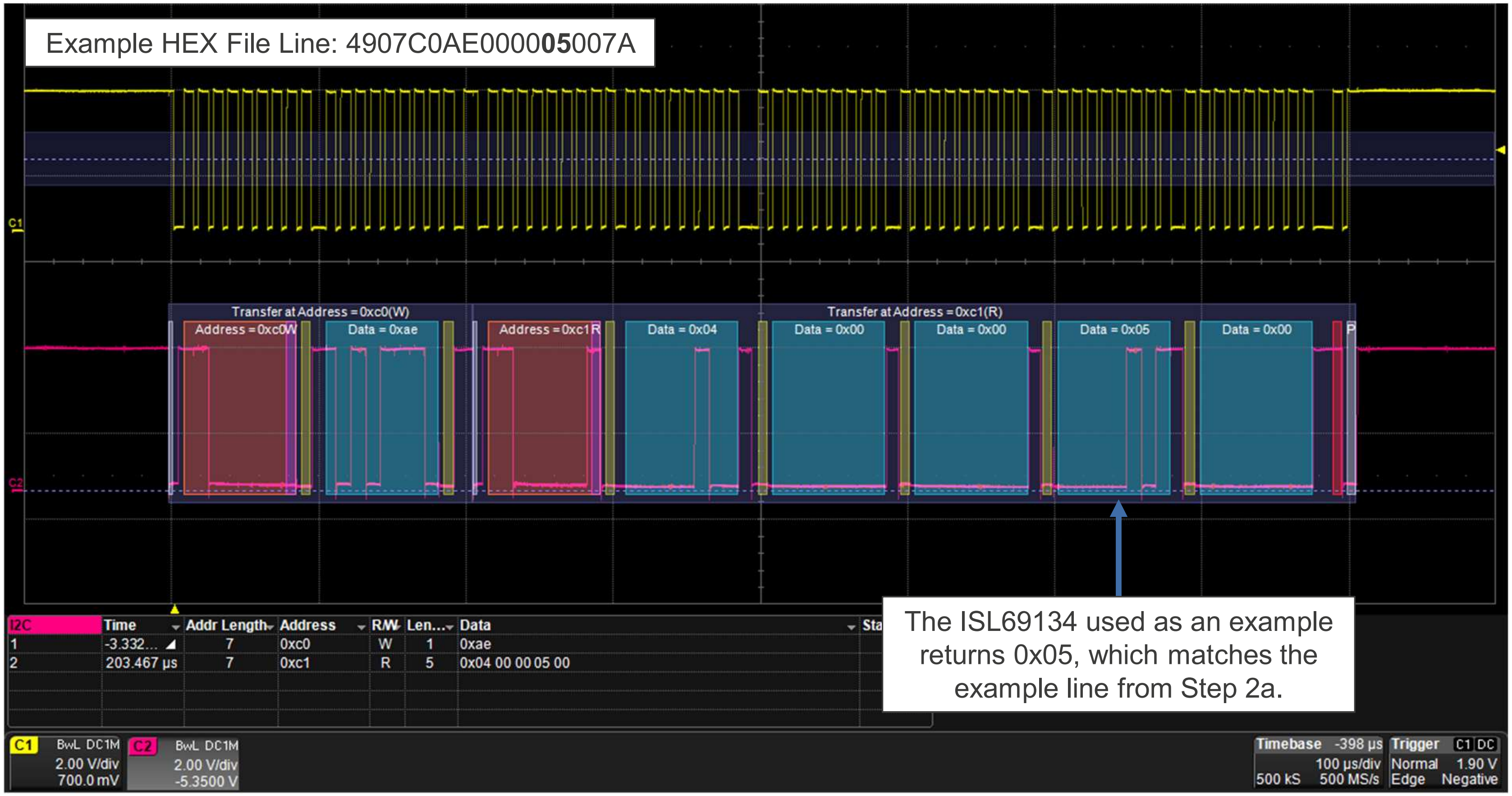
REV[0]	A	REV[1]	A	REV[2]	A	REV[3]	N	P
0xXX		0xXX		0xYY		0xXX		

For the purposes of programming devices, these bytes can be ignored.

This byte must match the byte in the same position in the HEX file IC_DEVICE_REV line.

For the purposes of programming devices, this byte can be ignored.

Step 2c – Example Waveforms



Example HEX File Line: 4907C0AE0000**05**007A

I2C	Time	Addr Length	Address	R/W	Len...	Data	Sta
1	-3.332...	7	0xc0	W	1	0xae	
2	203.467 µs	7	0xc1	R	5	0x04 00 00 05 00	

C1 BwL DC1M 2.00 V/div 700.0 mV

C2 BwL DC1M 2.00 V/div -5.3500 V

Timebase -398 µs 100 µs/div 500 kS 500 MS/s

Trigger C1 DC Normal 1.90 V Edge Negative

The ISL69134 used as an example returns 0x05, which matches the example line from Step 2a.

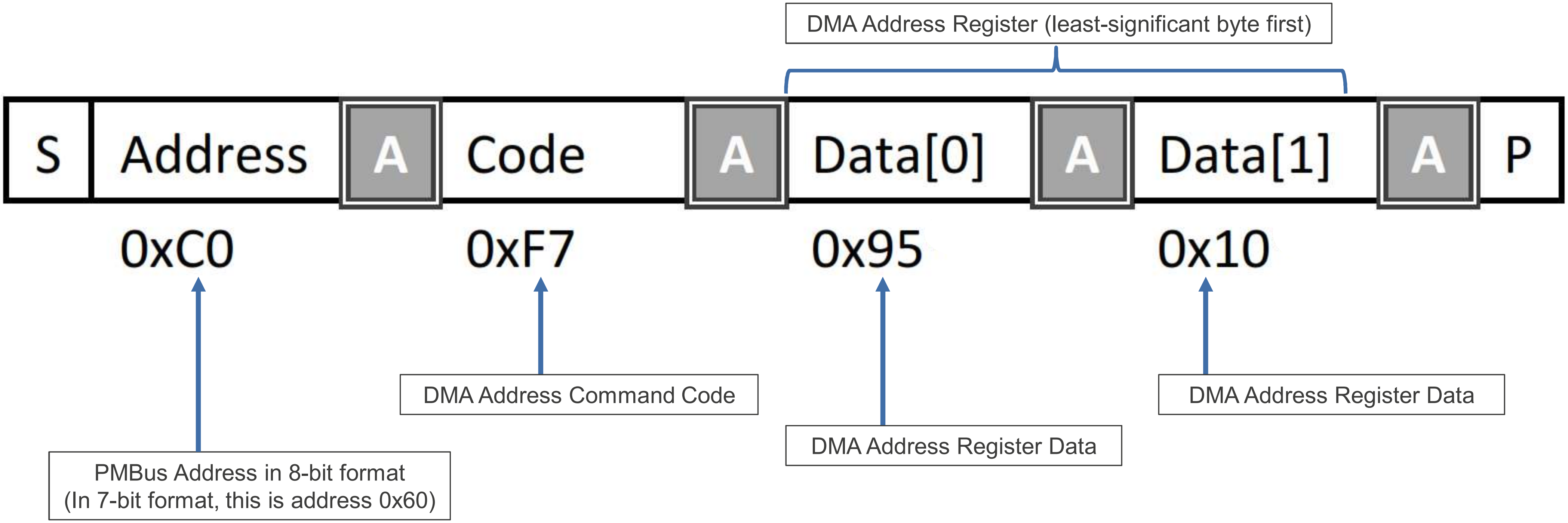
Step 3 – Enable Full Power Mode or Regulation

- To program an already programmed part:
  a. If neither rail is enabled, enable full power mode on the part before programming.
  b. If one or both rails are enabled, allow the part to regulate voltage while programming.

- **Only perform one of the above options. Do not perform both.**

Step 3a – Enable Full Power Mode

To enable full power mode, first write to the DMA address register as shown below then read the DMA data register.



DMA Address Register (least-significant byte first)

S	Address	A	Code	A	Data[0]	A	Data[1]	A	P
	0xC0		0xF7		0x95		0x10		

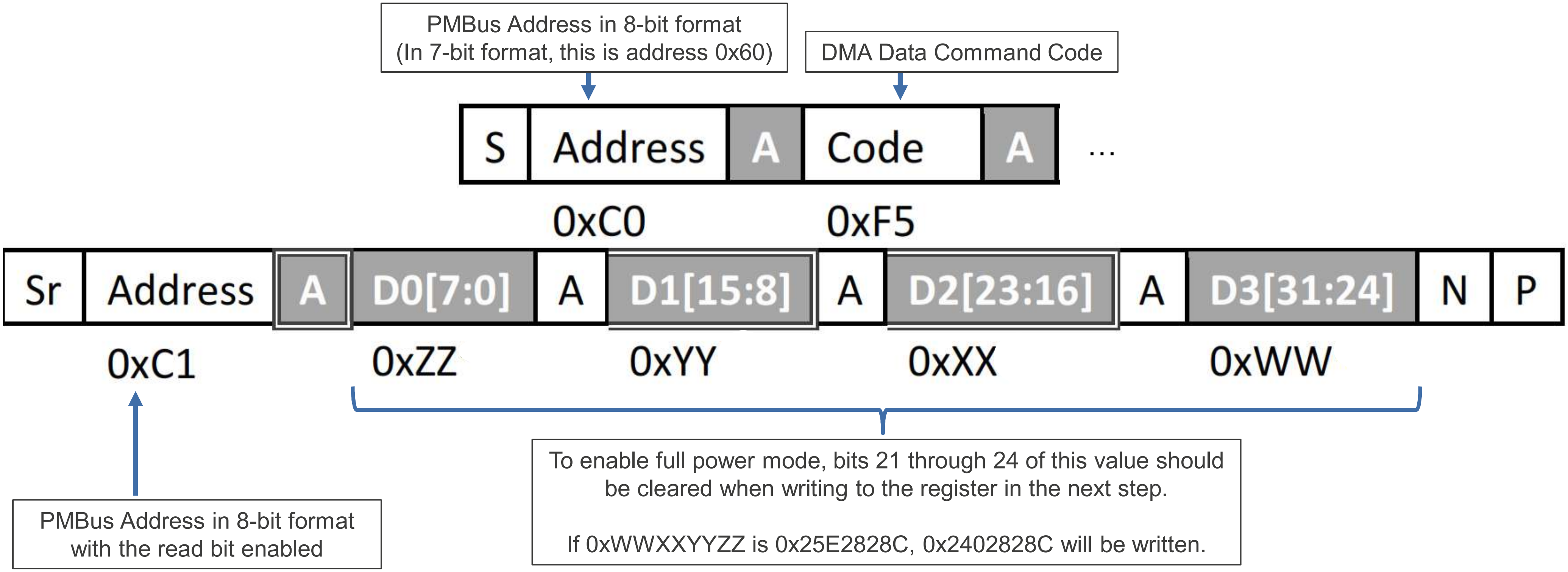
DMA Address Command Code

DMA Address Register Data

DMA Address Register Data

PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

Step 3a – Enable Full Power Mode



PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

DMA Data Command Code

S | Address | A | Code | A …

0xC0 0xF5

Sr | Address | A | D0[7:0] | A | D1[15:8] | A | D2[23:16] | A | D3[31:24] | N | P

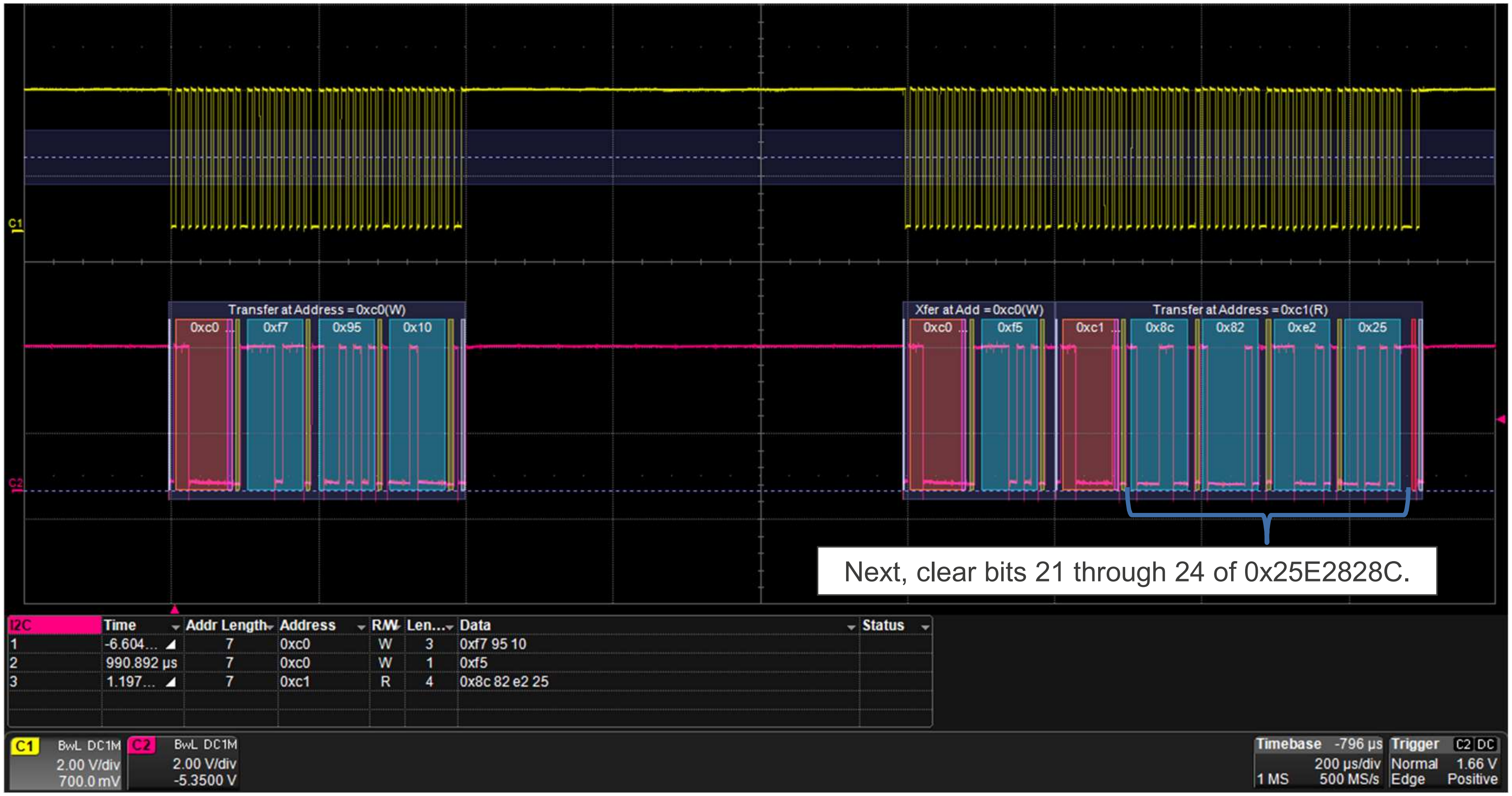
0xC1 0xZZ 0xYY 0xXX 0xWW

PMBus Address in 8-bit format
with the read bit enabled

To enable full power mode, bits 21 through 24 of this value should be cleared when writing to the register in the next step.

If 0xWWXXYYZZ is 0x25E2828C, 0x2402828C will be written.

Step 3a – Example Waveforms



Next, clear bits 21 through 24 of 0x25E2828C.

I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-6.604...	7	0xc0	W	3	0xf7 95 10	
2	990.892 µs	7	0xc0	W	1	0xf5	
3	1.197...	7	0xc1	R	4	0x8c 82 e2 25	

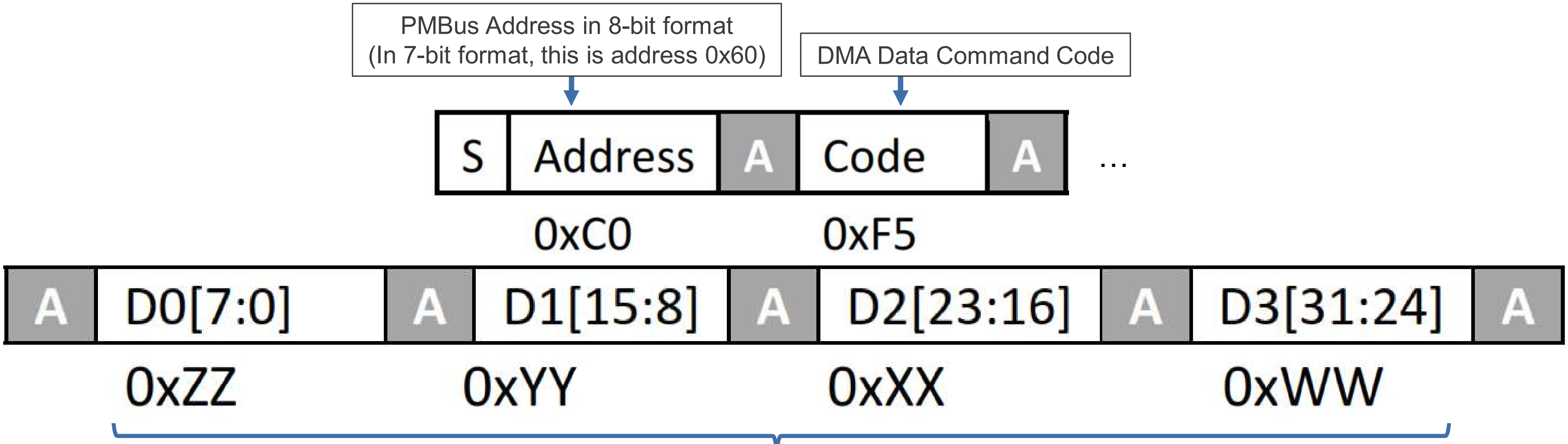
C1 BwL DC1M 2.00 V/div 700.0 mV

C2 BwL DC1M 2.00 V/div -5.3500 V

Timebase -796 µs 1 MS 200 µs/div 500 MS/s

Trigger C2 DC Normal 1.66 V Edge Positive

Step 3a – Enable Full Power Mode

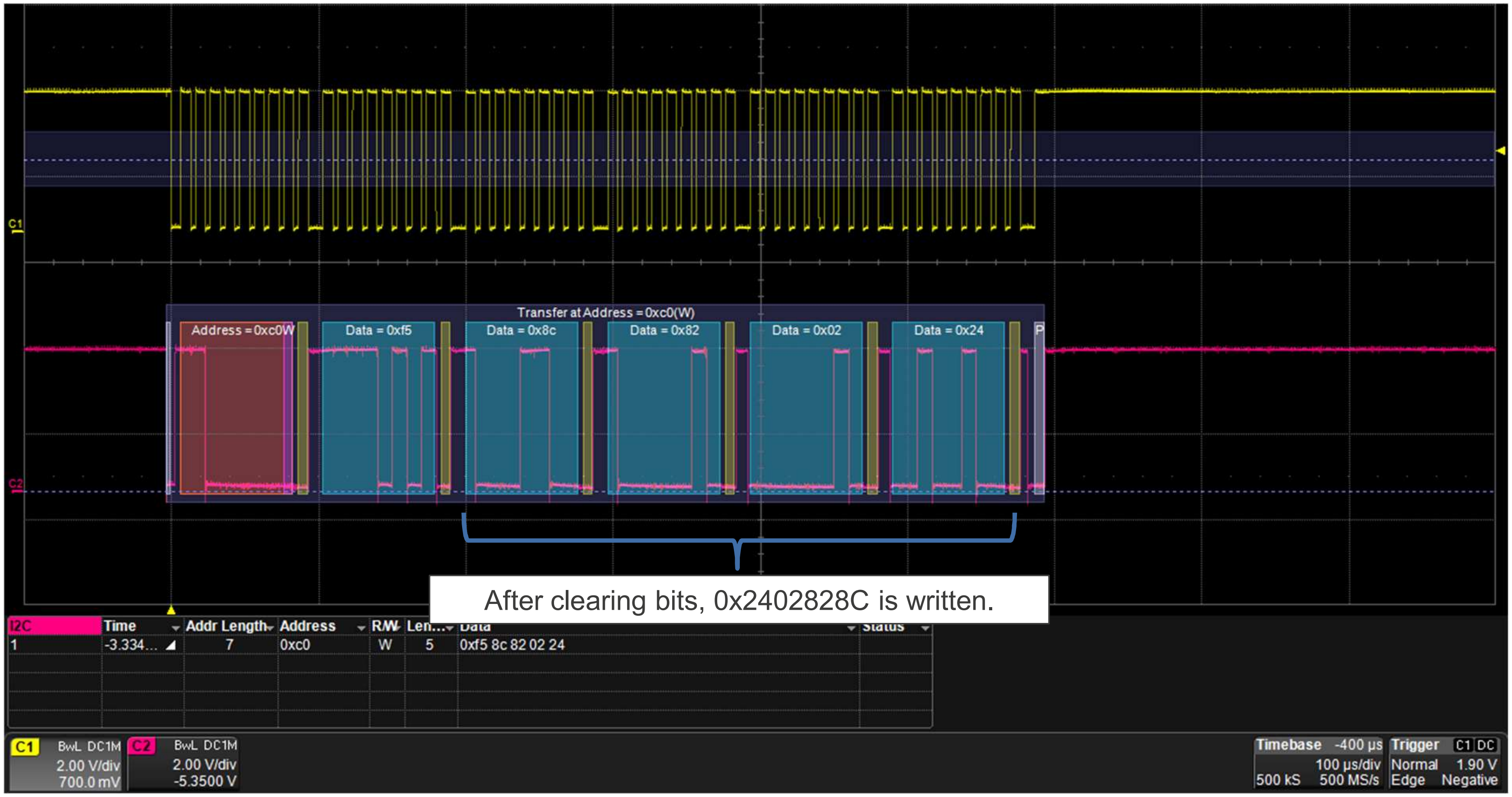


PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

DMA Data Command Code

This data should be the 8-byte data from the previous page. In the example, this was 0x2402828C. To write in the example, set the following:
0xWW = 0x24
0xXX = 0x02
0xYY = 0x82
0xZZ = 0x8C

Step 3a – Example Waveforms



After clearing bits, 0x2402828C is written.

I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-3.334...	7	0xc0	W	5	0xf5 8c 82 02 24	

C1 BwL DC1M 2.00 V/div 700.0 mV

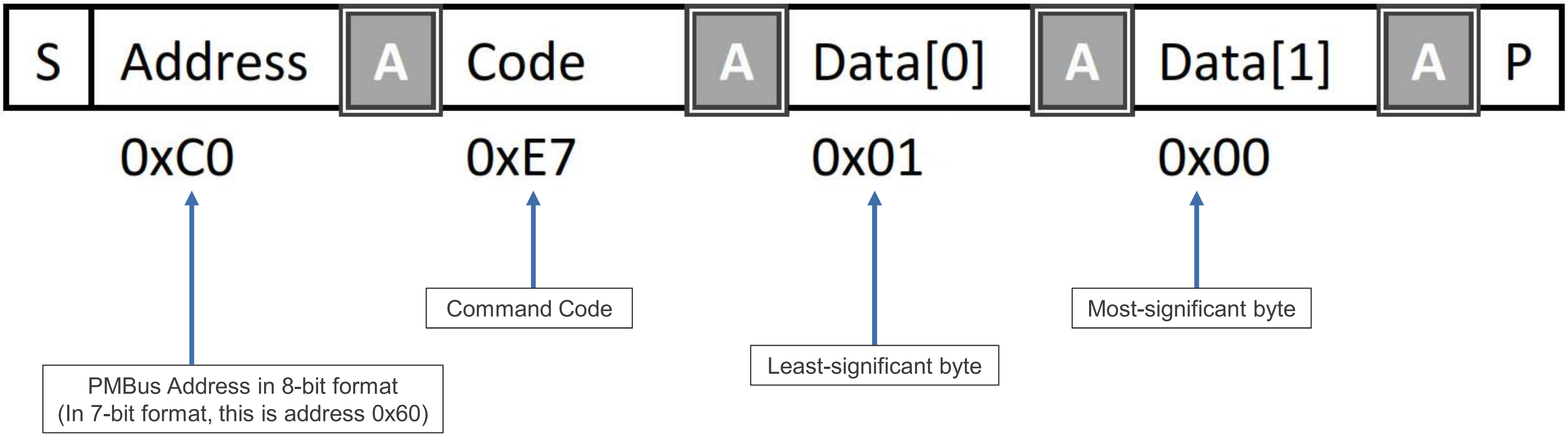
C2 BwL DC1M 2.00 V/div -5.3500 V

Timebase -400 µs 100 µs/div 500 kS 500 MS/s

Trigger C1 DC Normal 1.90 V Edge Negative

Step 3a – Enable Full Power Mode

To finish enabling full power mode, write 0x0001 to the device as shown below.



S	Address	A	Code	A	Data[0]	A	Data[1]	A	P
	0xC0		0xE7		0x01		0x00		

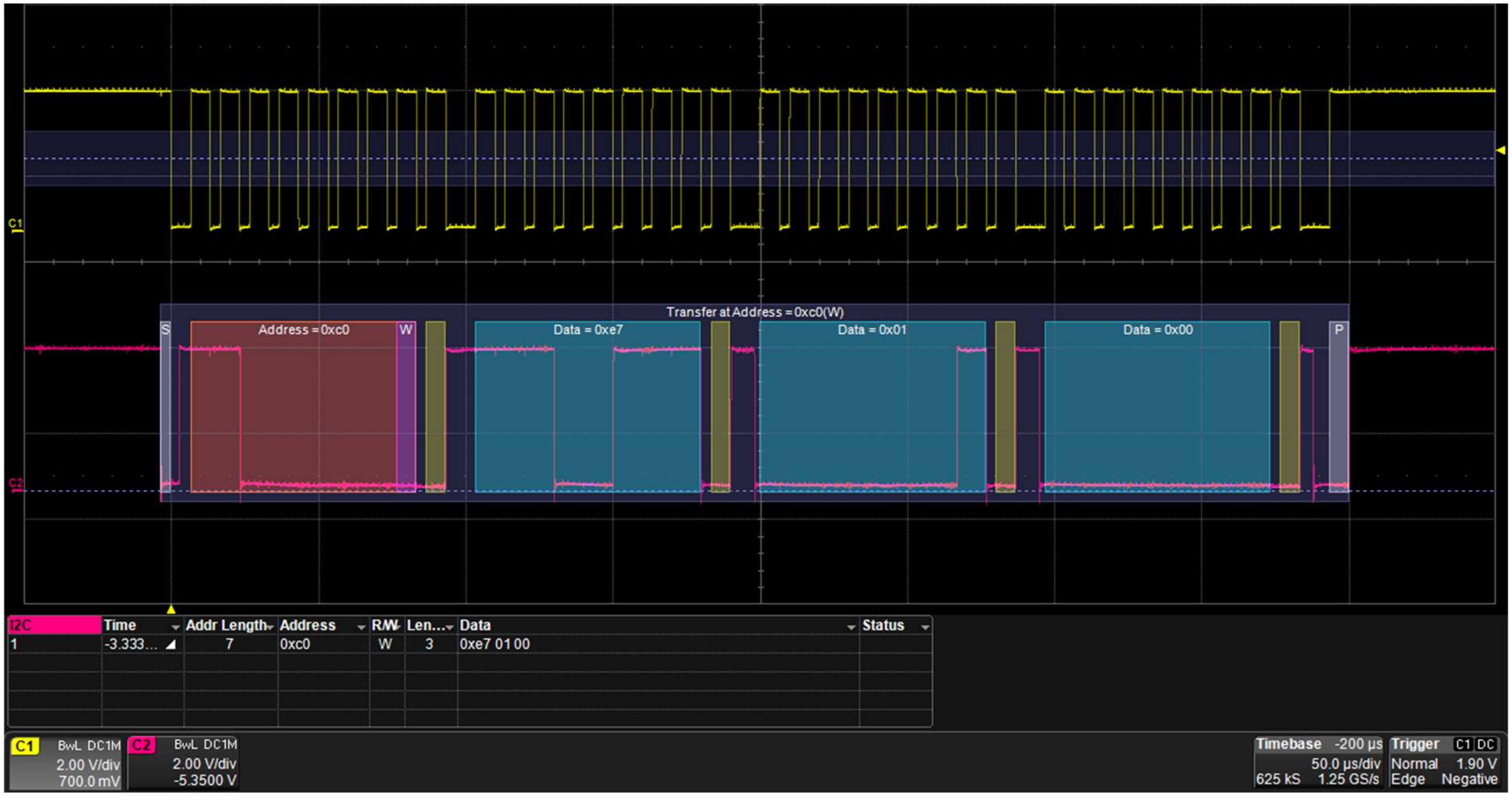
PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

Command Code

Least-significant byte

Most-significant byte

Step 3a – Example Waveforms



I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-3.333...	7	0xc0	W	3	0xe7 01 00	

C1 BwL DC1M 2.00 V/div 700.0 mV

C2 BwL DC1M 2.00 V/div -5.3500 V

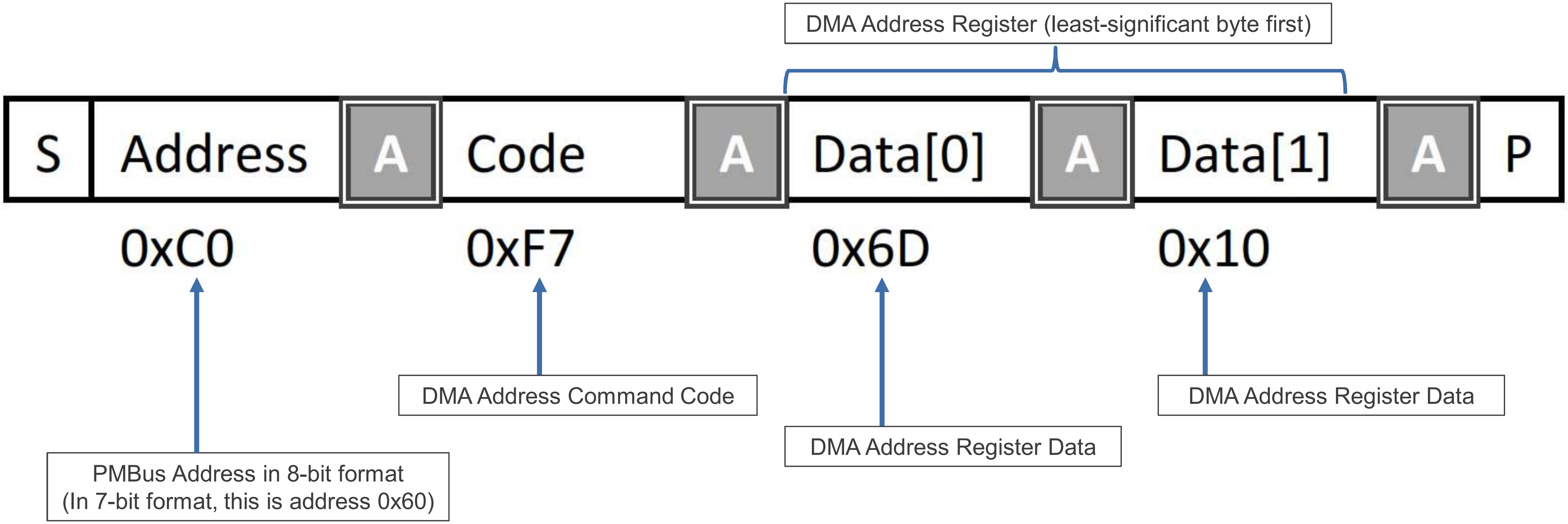
Timebase -200 µs 50.0 µs/div 625 kS 1.25 GS/s

Trigger C1 DC Normal 1.90 V Edge Negative

Step 3b – Enable Voltage Regulation

First write to the DMA address register as shown below then read the DMA data register.

If Step 3a was completed, go directly to Step 4. Do not complete Step 3b.



DMA Address Register (least-significant byte first)

S	Address	A	Code	A	Data[0]	A	Data[1]	A	P
	0xC0		0xF7		0x6D		0x10		

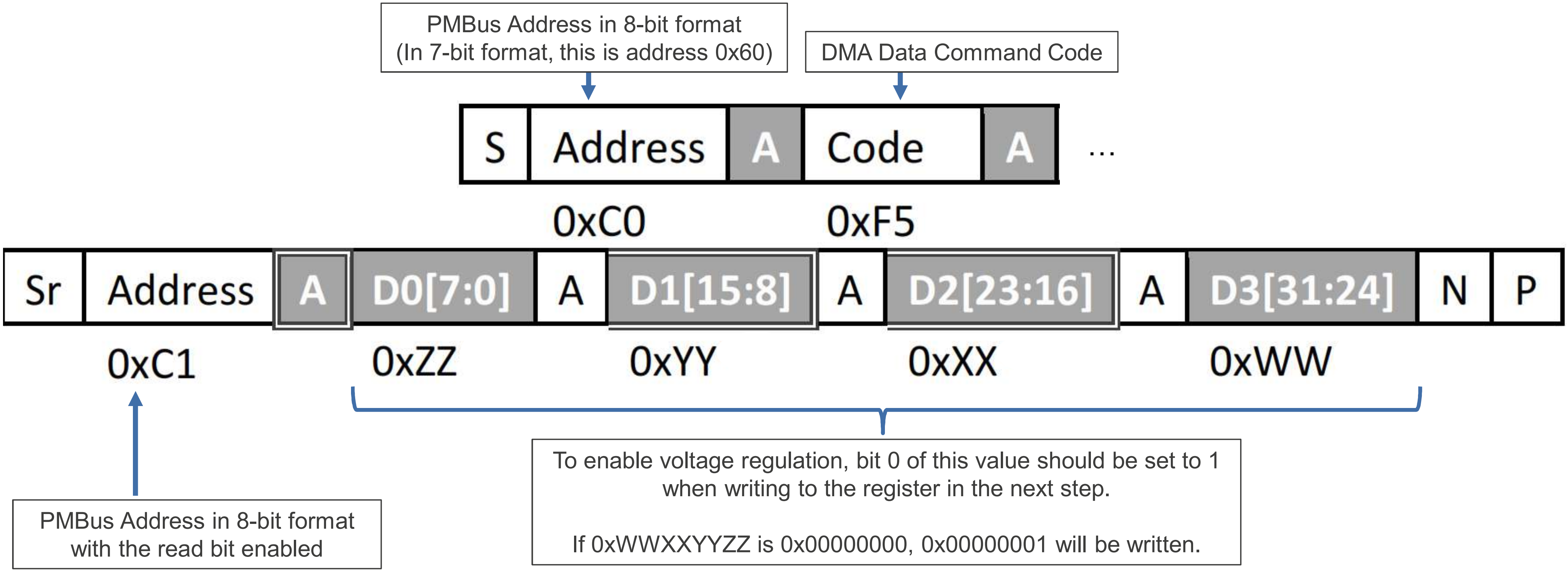
PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

DMA Address Command Code

DMA Address Register Data

DMA Address Register Data

Step 3b – Enable Voltage Regulation



PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

DMA Data Command Code

S	Address	A	Code	A	…
	0xC0		0xF5		

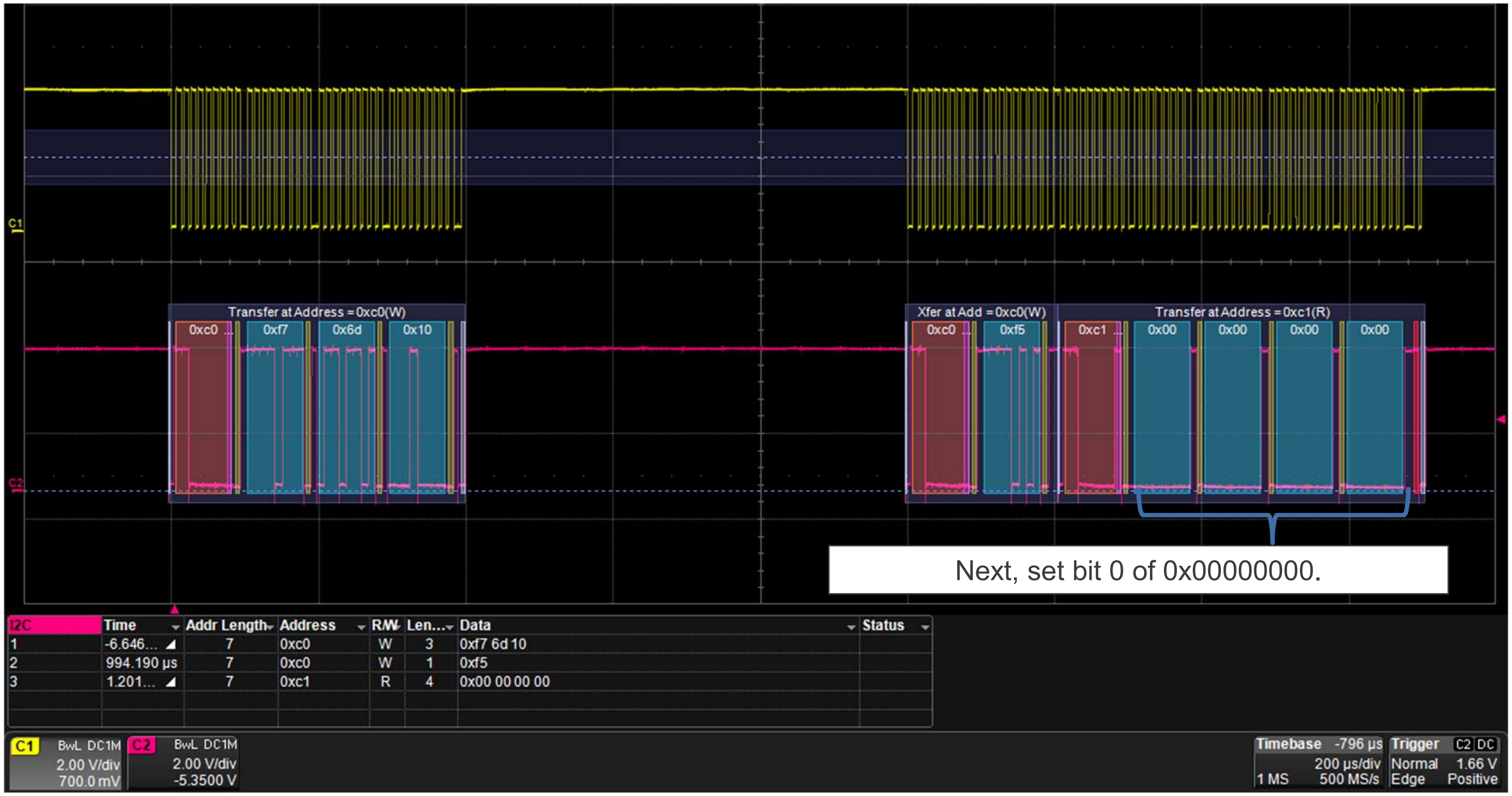
Sr	Address	A	D0[7:0]	A	D1[15:8]	A	D2[23:16]	A	D3[31:24]	N	P
	0xC1		0xZZ		0xYY		0xXX		0xWW		

PMBus Address in 8-bit format
with the read bit enabled

To enable voltage regulation, bit 0 of this value should be set to 1 when writing to the register in the next step.

If 0xWWXXYYZZ is 0x00000000, 0x00000001 will be written.

Step 3b – Example Waveforms



Next, set bit 0 of 0x00000000.

I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-6.646...	7	0xc0	W	3	0xf7 6d 10	
2	994.190 µs	7	0xc0	W	1	0xf5	
3	1.201...	7	0xc1	R	4	0x00 00 00 00	

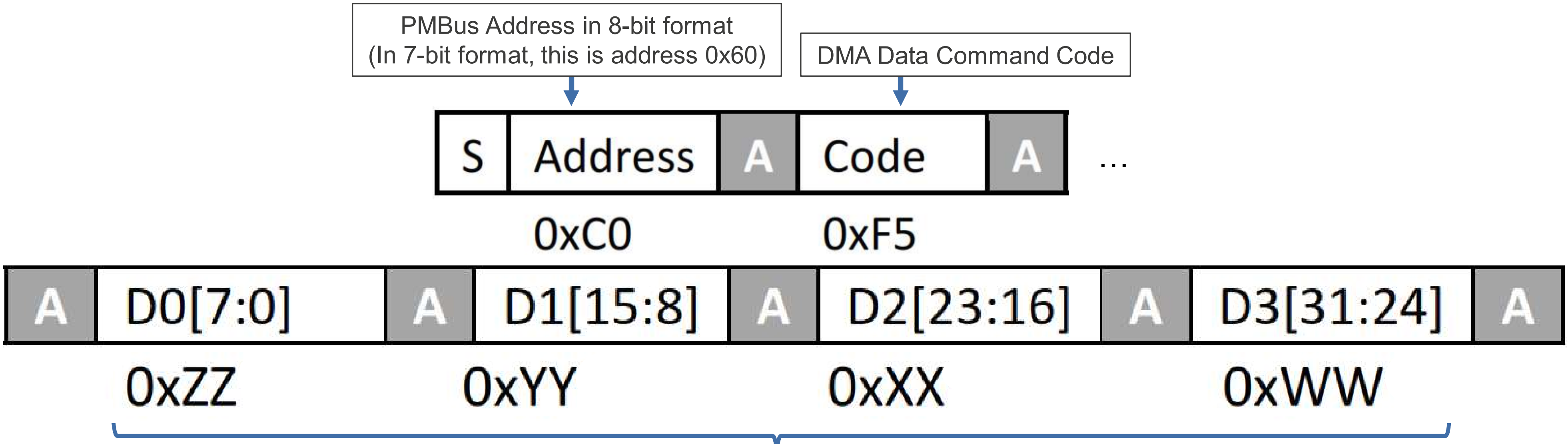
C1 BwL DC1M 2.00 V/div 700.0 mV

C2 BwL DC1M 2.00 V/div -5.3500 V

Timebase -796 µs 200 µs/div 1 MS 500 MS/s

Trigger C2 DC Normal 1.66 V Edge Positive

Step 3b – Enable Voltage Regulation



PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

DMA Data Command Code

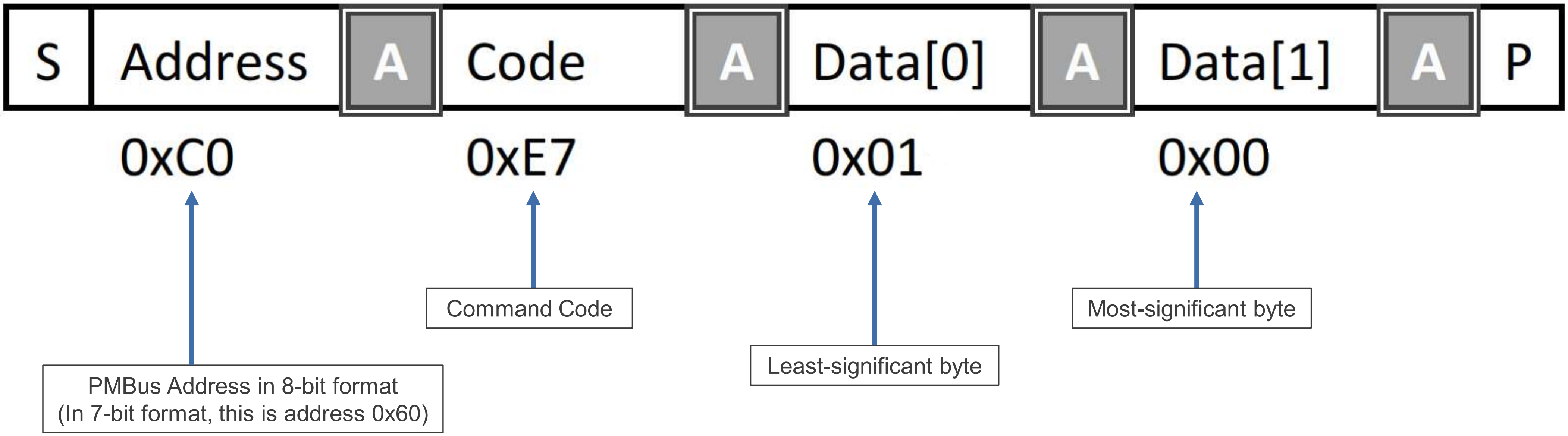
S	Address	A	Code	A	…
	0xC0		0xF5		

A	D0[7:0]	A	D1[15:8]	A	D2[23:16]	A	D3[31:24]	A
	0xZZ		0xYY		0xXX		0xWW	

This data should be the 8-byte data from the previous page. In the example, this was 0x00000001. To write in the example, set the following:
0xWW = 0x00
0xXX = 0x00
0xYY = 0x00
0xZZ = 0x01

Step 3b – Enable Voltage Regulation

To finish enabling voltage regulation, write 0x0001 to the device as shown below.



S	Address	A	Code	A	Data[0]	A	Data[1]	A	P
	0xC0		0xE7		0x01		0x00		

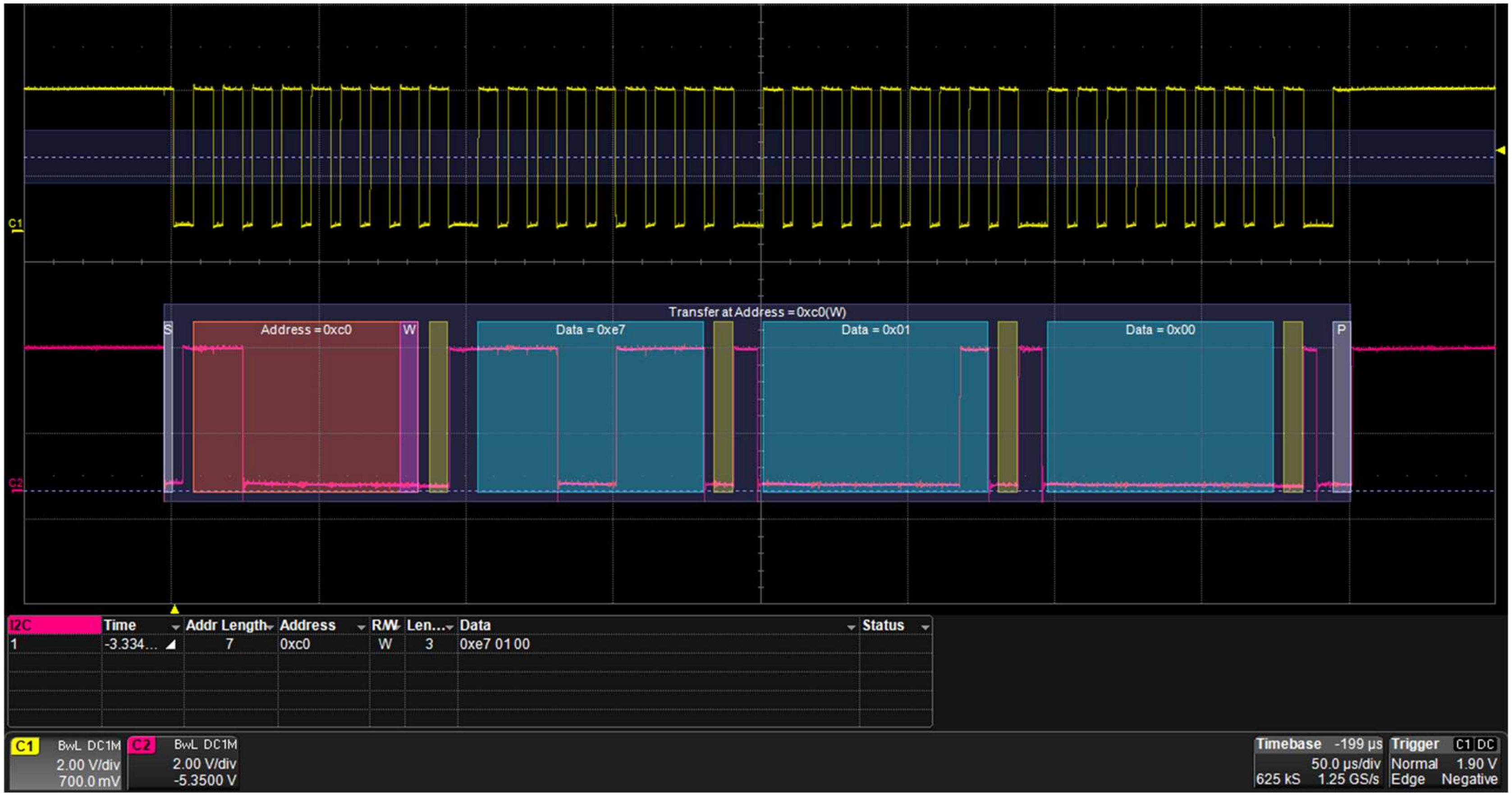
PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

Command Code

Least-significant byte

Most-significant byte

Step 3b – Example Waveforms



I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-3.334...	7	0xc0	W	3	0xe7 01 00	

C1 BwL DC1M 2.00 V/div 700.0 mV

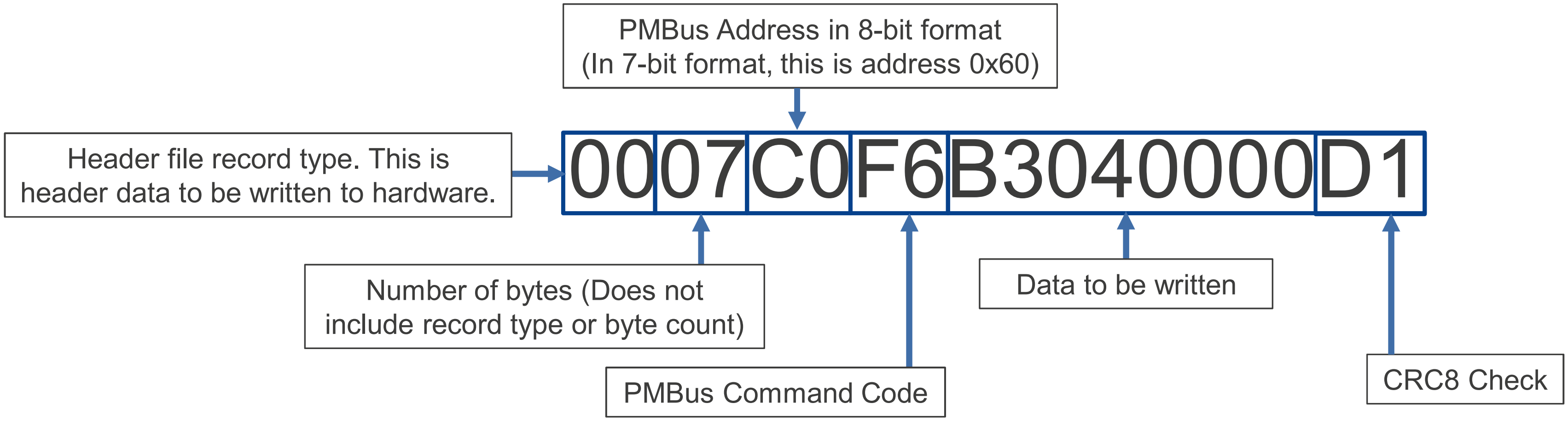
C2 BwL DC1M 2.00 V/div -5.3500 V

Timebase -199 µs 50.0 µs/div 625 kS 1.25 GS/s

Trigger C1 DC Normal 1.90 V Edge Negative

Step 4 – Parse HEX File and Write to Hardware

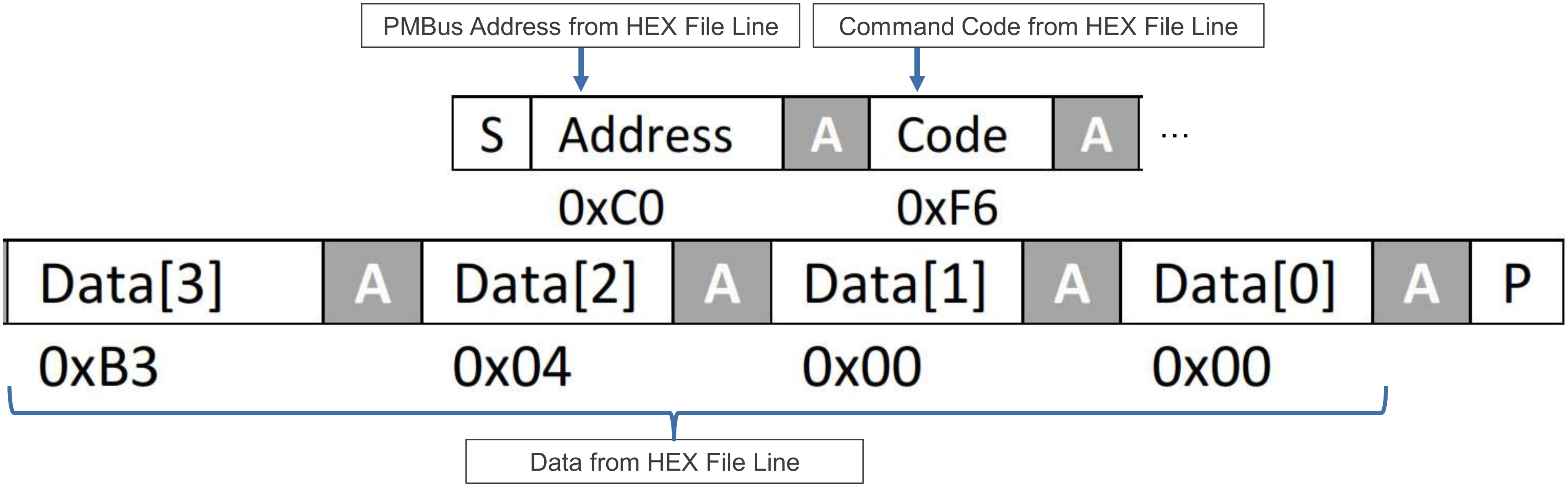
Example HEX File Line:


PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)
Header file record type. This is header data to be written to hardware.
00
07
C0
F6
B3040000
D1
Number of bytes (Does not include record type or byte count)
PMBus Command Code
Data to be written
CRC8 Check

Step 4 – Parse HEX File and Write to Hardware

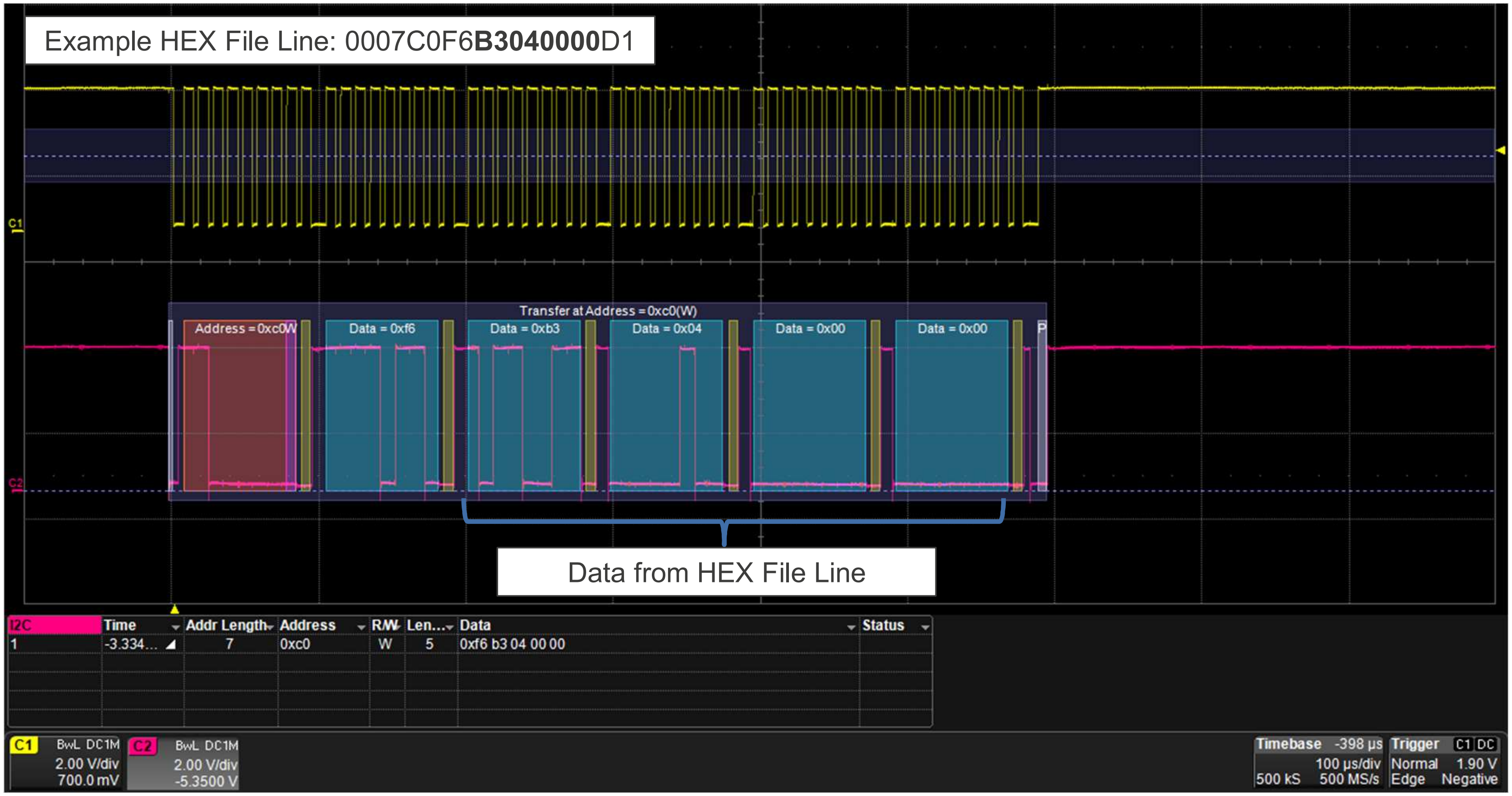
Example PMBus Command:

Hex File Line Data: 0007C0F6**B3040000**D1



This write must be performed with **ALL** lines in the HEX file that do not start with 0x49.

Step 4 – Example Waveforms



Example HEX File Line: 0007C0F6**B3040000**D1

Data from HEX File Line

I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-3.334...	7	0xc0	W	5	0xf6 b3 04 00 00	

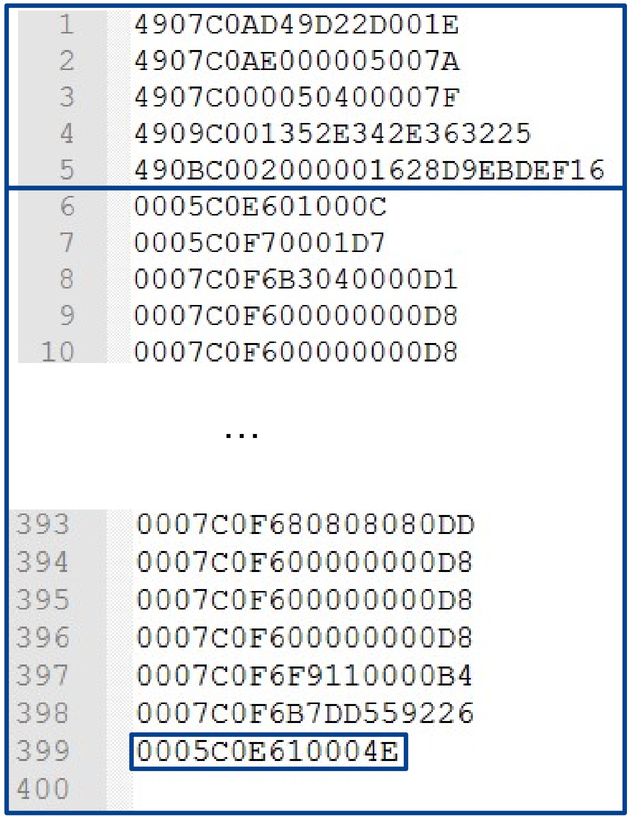
C1 BwL DC1M 2.00 V/div 700.0 mV

C2 BwL DC1M 2.00 V/div -5.3500 V

Timebase -398 µs 100 µs/div 500 kS 500 MS/s

Trigger C1 DC Normal 1.90 V Edge Negative

Step 4 – Example HEX File



```
4907C0AD49D22D001E
4907C0AE000005007A
4907C000050400007F
4909C001352E342E363225
490BC002000001628D9EBDEF16
0005C0E601000C
0005C0F70001D7
0007C0F6B3040000D1
0007C0F600000000D8
0007C0F600000000D8

        ...

0007C0F680808080DD
0007C0F600000000D8
0007C0F600000000D8
0007C0F600000000D8
0007C0F6F9110000B4
0007C0F6B7DD559226
0005C0E610004E
```

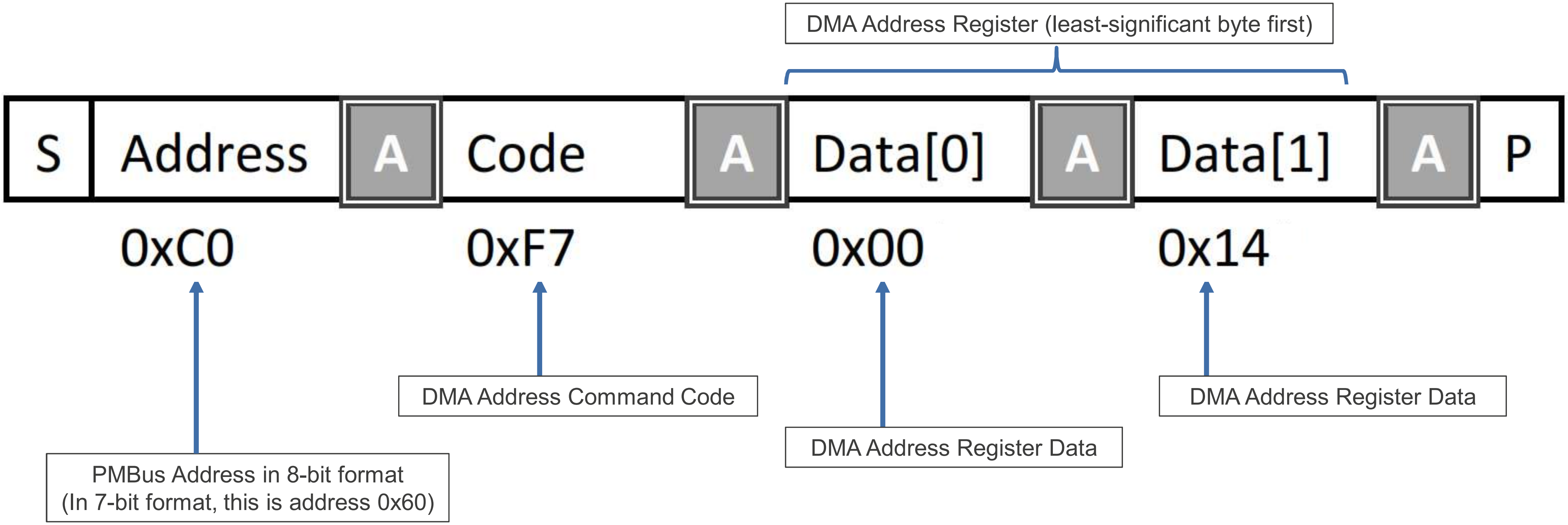
In Step 4, ignore these lines.

Complete Step 4 for all remaining lines.

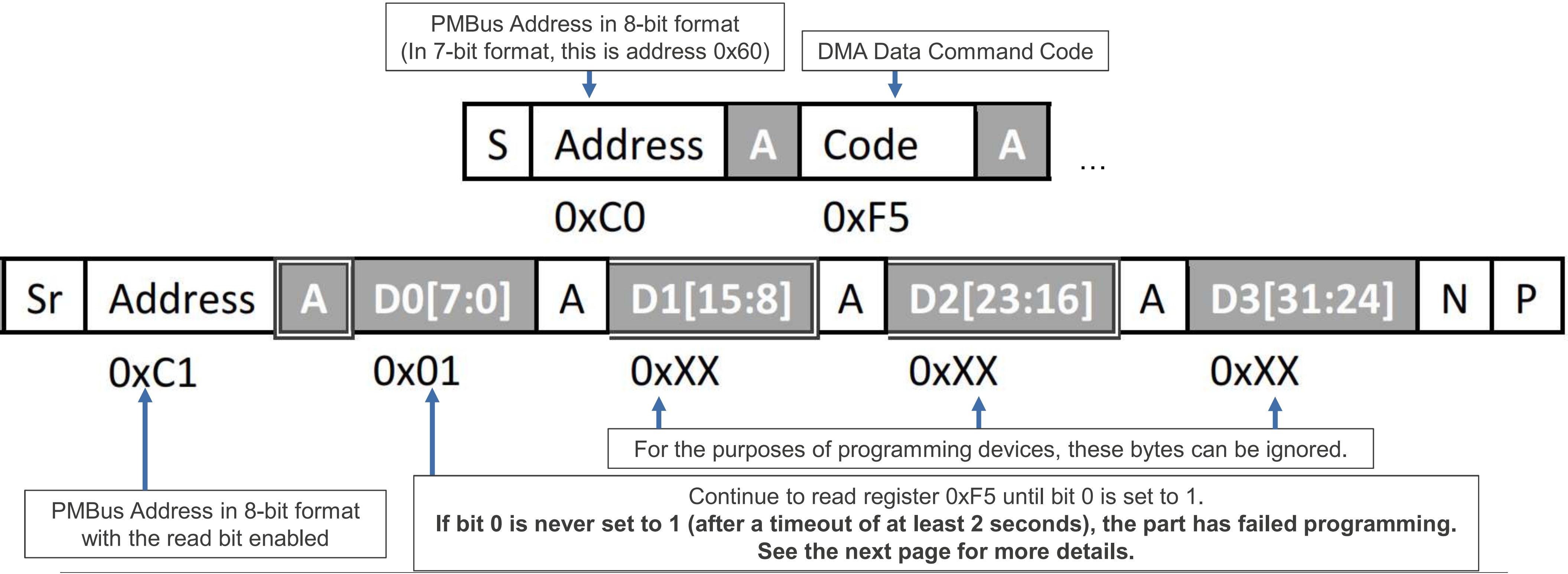
Complete Step 4 for the end of file line.

Step 5a – Poll PROGRAMMER_STATUS Register

To check for the completion of device programming, first write to the DMA address register as shown below then read the DMA data register until bit 0 is set to 1.


DMA Address Register (least-significant byte first)
S
Address
A
Code
A
Data[0]
A
Data[1]
A
P
0xC0
0xF7
0x00
0x14
DMA Address Command Code
DMA Address Register Data
DMA Address Register Data
PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

Step 5a – Poll PROGRAMMER_STATUS Register



PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

DMA Data Command Code

S	Address	A	Code	A	…
	0xC0		0xF5		

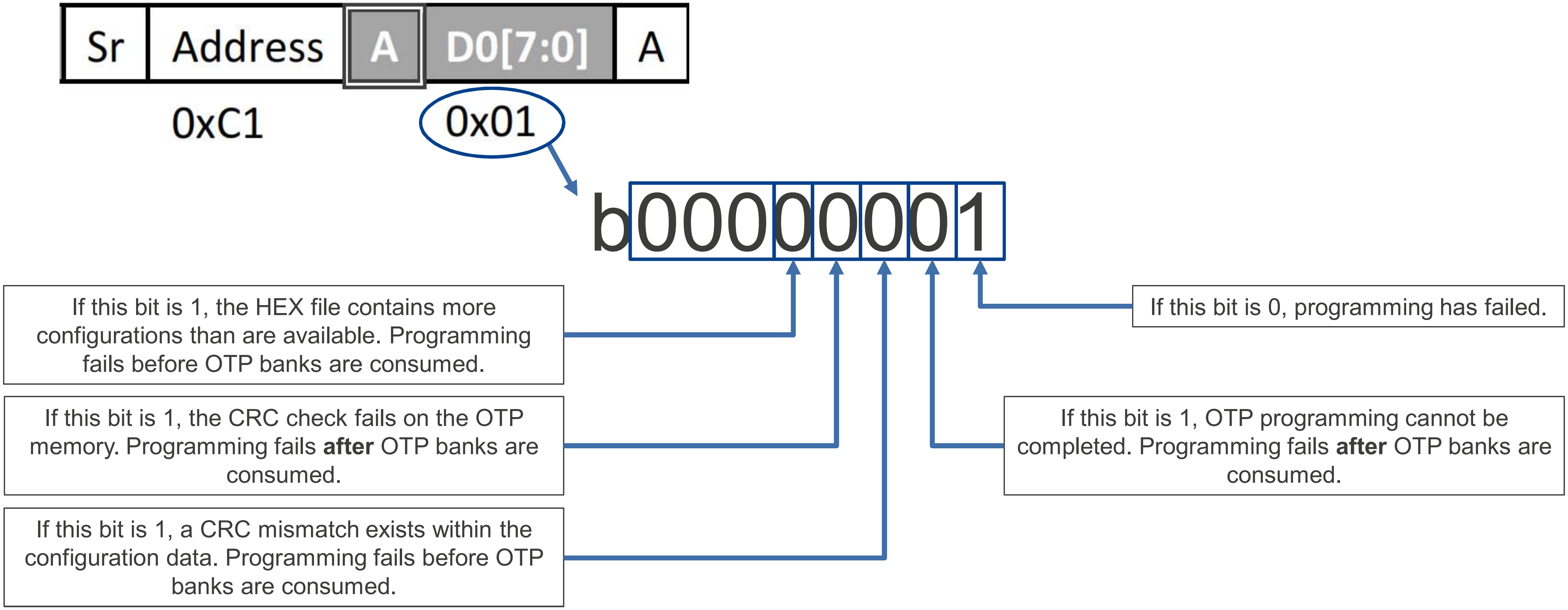
Sr	Address	A	D0[7:0]	A	D1[15:8]	A	D2[23:16]	A	D3[31:24]	N	P
	0xC1		0x01		0xXX		0xXX		0xXX		

For the purposes of programming devices, these bytes can be ignored.

PMBus Address in 8-bit format
with the read bit enabled

Continue to read register 0xF5 until bit 0 is set to 1.
If bit 0 is never set to 1 (after a timeout of at least 2 seconds), the part has failed programming.
See the next page for more details.

Step 5a – Programming Failure



If this bit is 1, the HEX file contains more configurations than are available. Programming fails before OTP banks are consumed.

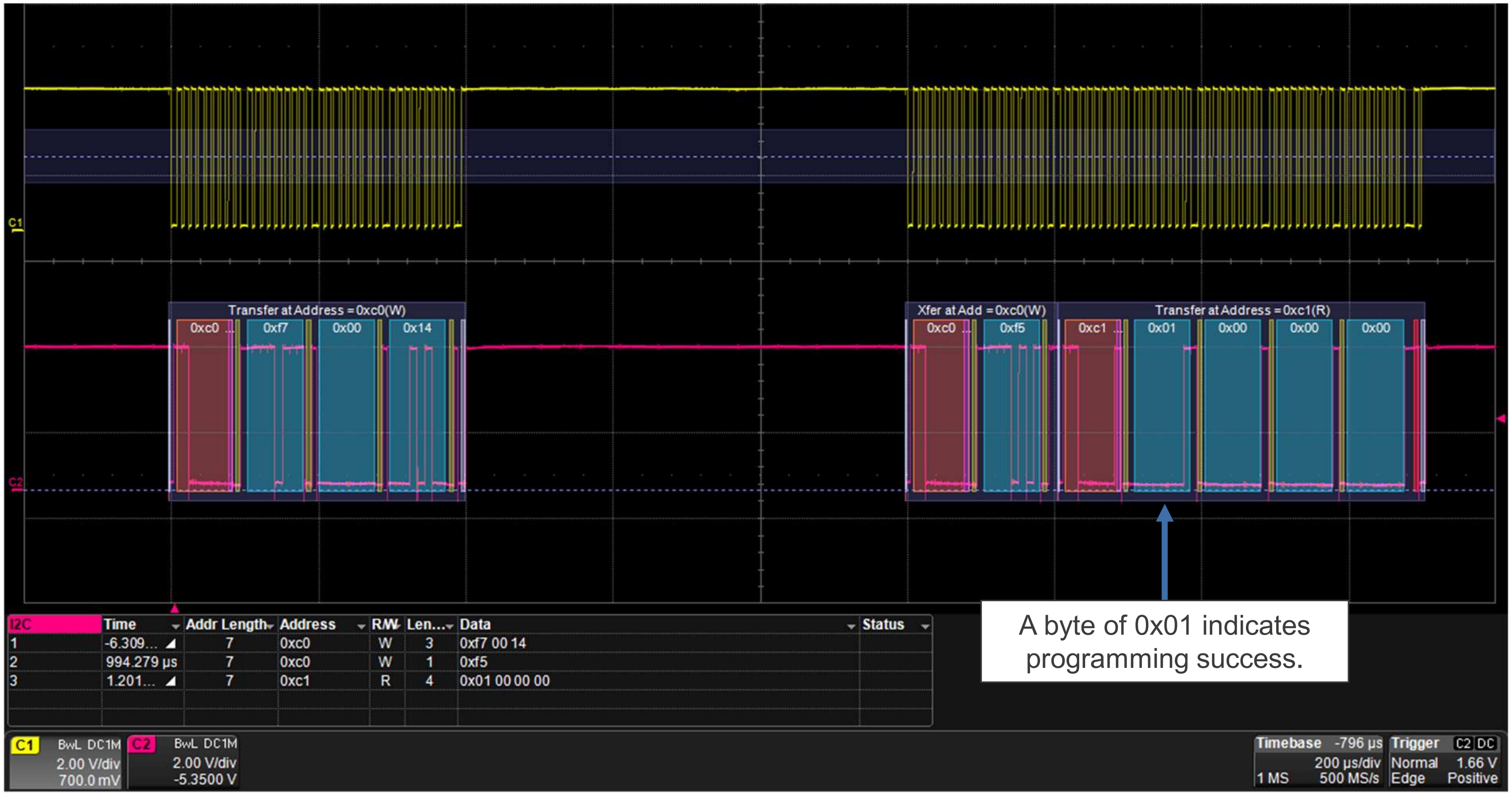
If this bit is 0, programming has failed.

If this bit is 1, the CRC check fails on the OTP memory. Programming fails **after** OTP banks are consumed.

If this bit is 1, OTP programming cannot be completed. Programming fails **after** OTP banks are consumed.

If this bit is 1, a CRC mismatch exists within the configuration data. Programming fails before OTP banks are consumed.

Step 5a – Example Waveforms



I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-6.309...	7	0xc0	W	3	0xf7 00 14	
2	994.279 µs	7	0xc0	W	1	0xf5	
3	1.201...	7	0xc1	R	4	0x01 00 00 00	

C1 BwL DC1M 2.00 V/div 700.0 mV

C2 BwL DC1M 2.00 V/div -5.3500 V

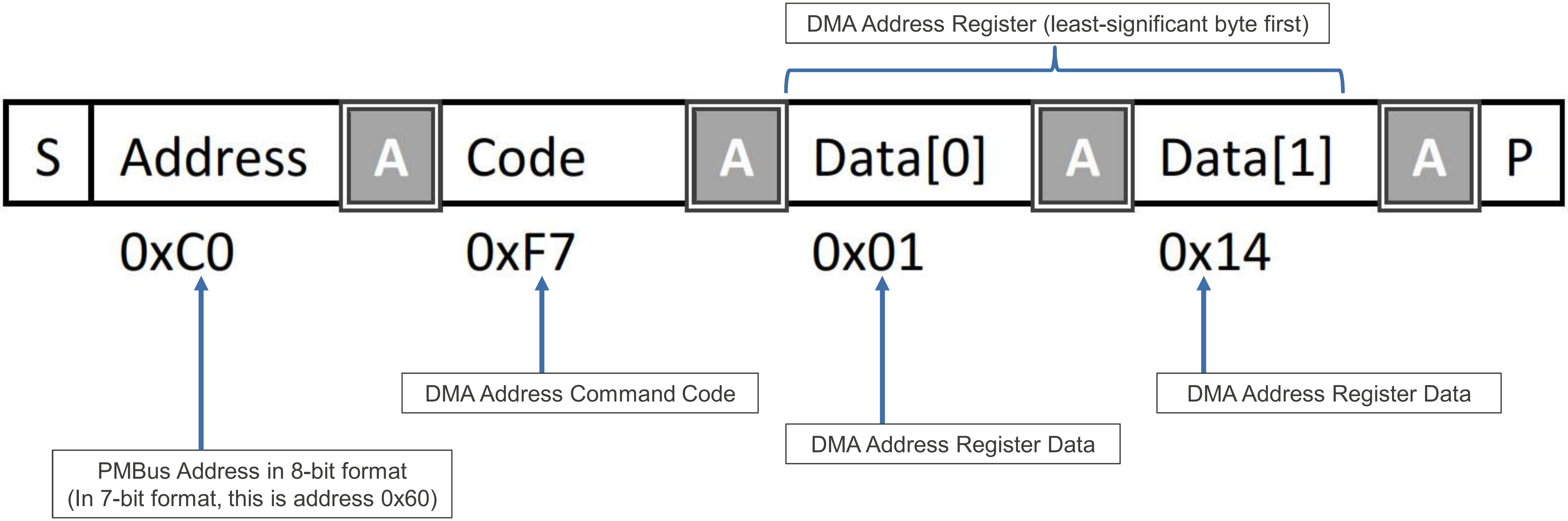
Timebase -796 µs 200 µs/div 1 MS 500 MS/s

Trigger C2 DC Normal 1.66 V Edge Positive

A byte of 0x01 indicates programming success.

Step 5b – Read BANK_STATUS Register

To read the BANK_STATUS register, first write to the DMA address register as shown below then read the DMA data register to retrieve BANK_STATUS data.



DMA Address Register (least-significant byte first)

S	Address	A	Code	A	Data[0]	A	Data[1]	A	P
	0xC0		0xF7		0x01		0x14		

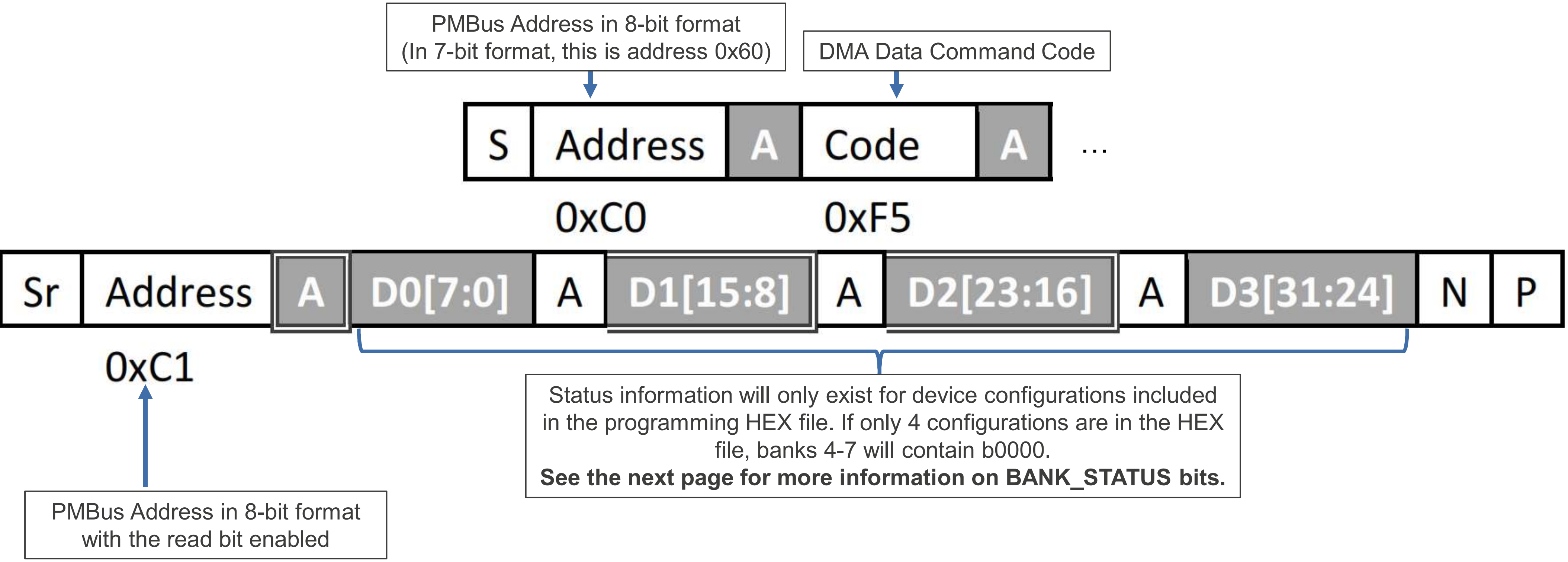
PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

DMA Address Command Code

DMA Address Register Data

DMA Address Register Data

Step 5b – Read BANK_STATUS Register



PMBus Address in 8-bit format
(In 7-bit format, this is address 0x60)

DMA Data Command Code

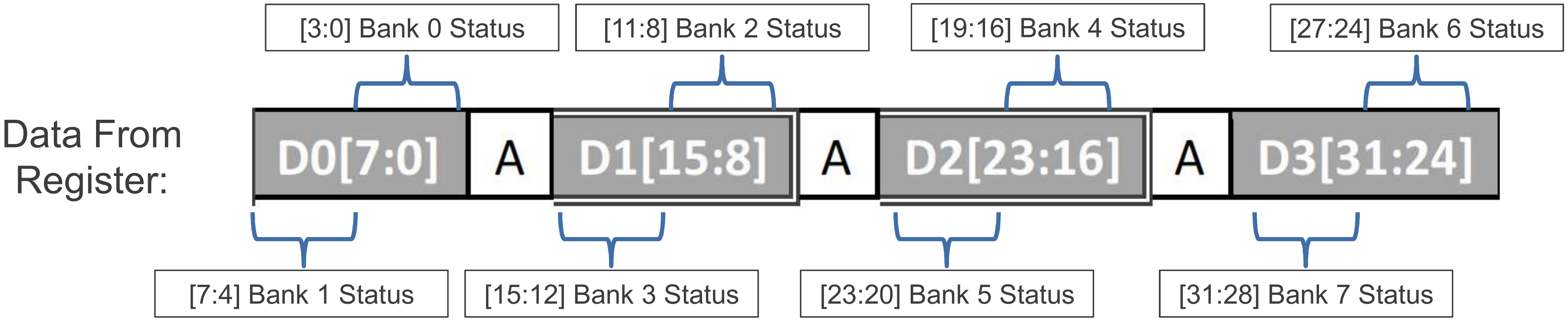
Status information will only exist for device configurations included in the programming HEX file. If only 4 configurations are in the HEX file, banks 4-7 will contain b0000.
See the next page for more information on BANK_STATUS bits.

PMBus Address in 8-bit format
with the read bit enabled

Step 5b – Read BANK_STATUS Register

Every set of bank status bits will correspond to the table below:

Bank Status Bits	Description
b1000	Fail: CRC mismatch OTP
b0100	Fail: CRC mismatch RAM
b0010	Fail: Write error
b0001	Bank Written
b0000	Bank Unaffected



Data From Register:

[3:0] Bank 0 Status

[11:8] Bank 2 Status

[19:16] Bank 4 Status

[27:24] Bank 6 Status

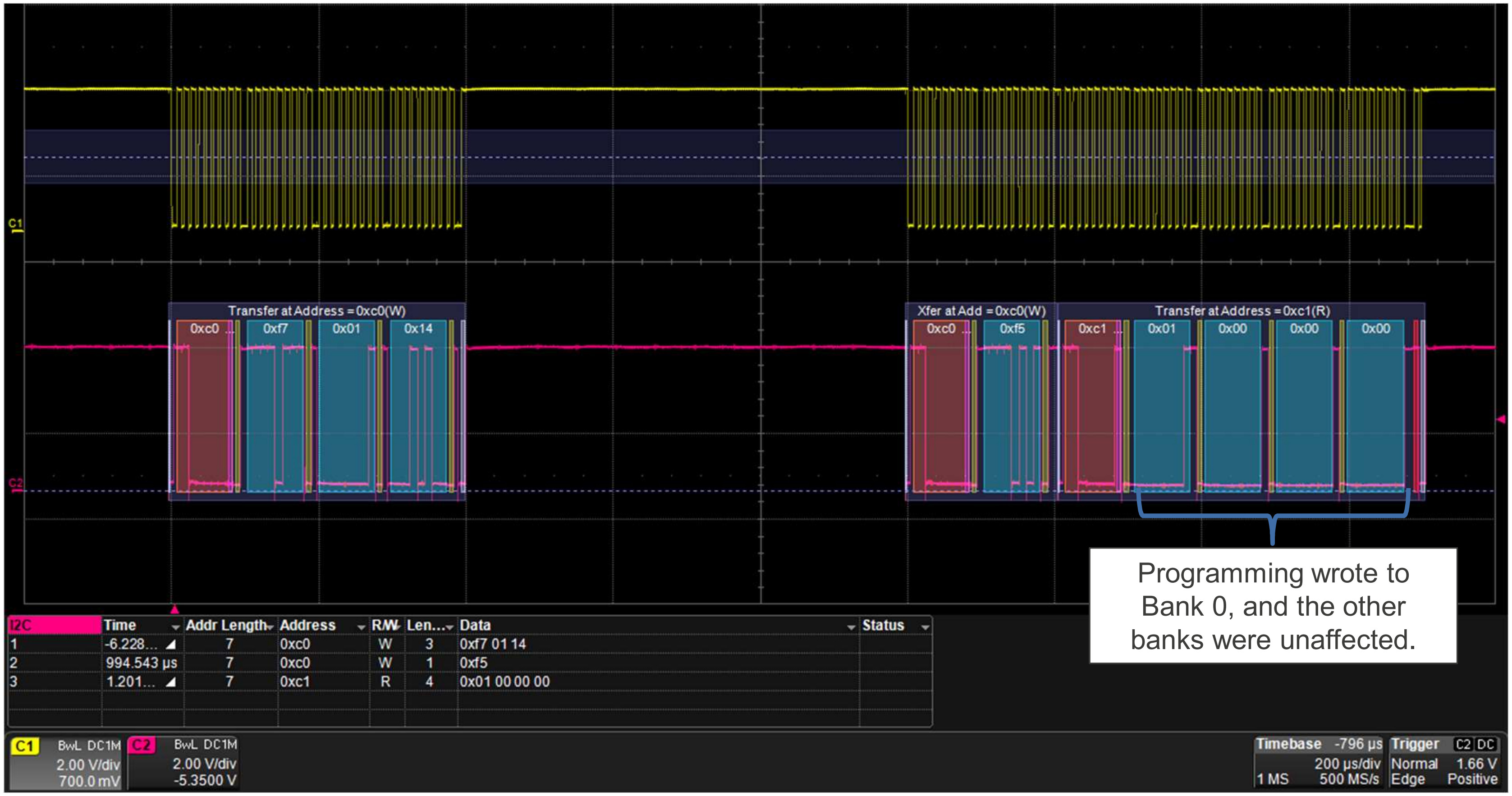
[7:4] Bank 1 Status

[15:12] Bank 3 Status

[23:20] Bank 5 Status

[31:28] Bank 7 Status

Step 5b – Example Waveforms



I2C	Time	Addr Length	Address	R/W	Len...	Data	Status
1	-6.228...	7	0xc0	W	3	0xf7 01 14	
2	994.543 µs	7	0xc0	W	1	0xf5	
3	1.201...	7	0xc1	R	4	0x01 00 00 00	

C1 BwL DC1M 2.00 V/div 700.0 mV

C2 BwL DC1M 2.00 V/div -5.3500 V

Timebase -796 µs 200 µs/div 1 MS 500 MS/s

Trigger C2 DC Normal 1.66 V Edge Positive

Programming wrote to Bank 0, and the other banks were unaffected.

Algorithm Completion

- After successfully completing programming, 3.3V VCC must be powered down to apply changes.

- Reading the new CRC from RAM is possible after cycling VCC with a 50ms delay.

BIG IDEAS FOR EVERY SPACE

Renesas.com