Basics of I2C: Reserved Addresses

TIPL 6103
TI Precision Labs – Digital Communications

Prepared by Joseph Wu

Presented by Alex Smith

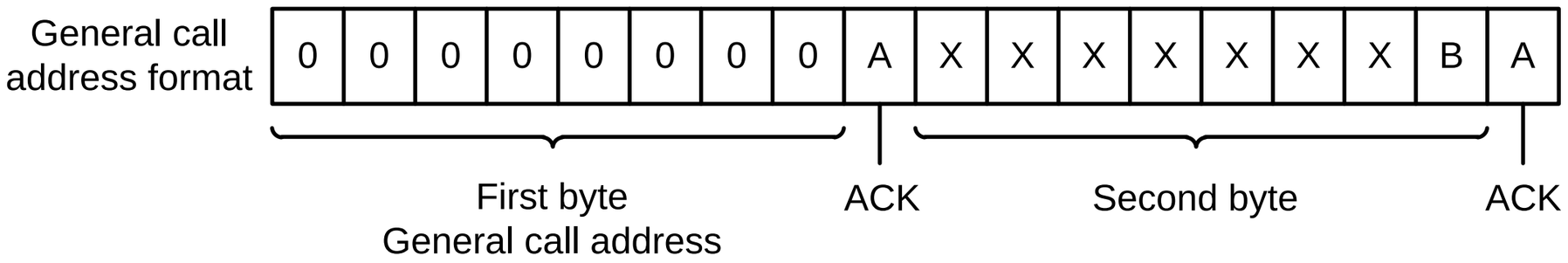
I2C Reserved Addresses

Target Address	R/$\overline{W}$ Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode controller code
111 11XX	1	Device ID
111 10XX	X	10-bit target addressing

I2C Reserved Addresses – General Call

Target Address	R/$\overline{W}$ Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode controller code
111 11XX	1	Device ID
111 10XX	X	10-bit target addressing

General call address
- Addressed as 0x00 write
- Communicates to all devices
- Second byte contains command
- Can be used for several functions including RESET



I2C Reserved Addresses – START Byte

Target Address	R/$\overline{W}$ Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode controller code
111 11XX	1	Device ID
111 10XX	X	10-bit target addressing

START Byte

- Addressed as 0x00 read
- Used for devices that may be polling the SDA and SCL lines instead of using an integrated I2C controller
- Devices can poll at a slower rate until it detects a 0 on the bus.
- The device can then switch to higher polling rates to detect the I2C transaction
- Once the I2C transaction is completed, the device can resume slower polling

I2C Reserved Addresses – Other Protocols

Target Address	R/$\overline{W}$ Bit	Description
0000 000	0	General call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Reserved for different bus format
0000 011	X	Reserved for future purposes
0000 1XX	X	Hs-mode controller code
1111 1XX	1	Device ID
1111 0XX	X	10-bit target addressing

CBUS address
- Address 0x01
- Ignored by I2C bus devices
- CBUS no longer in use

Reserved for different bus format
- Address 0x02
- Allows for mixed protocols
- Only I2C devices than can work with different formats and protocols may respond to this address

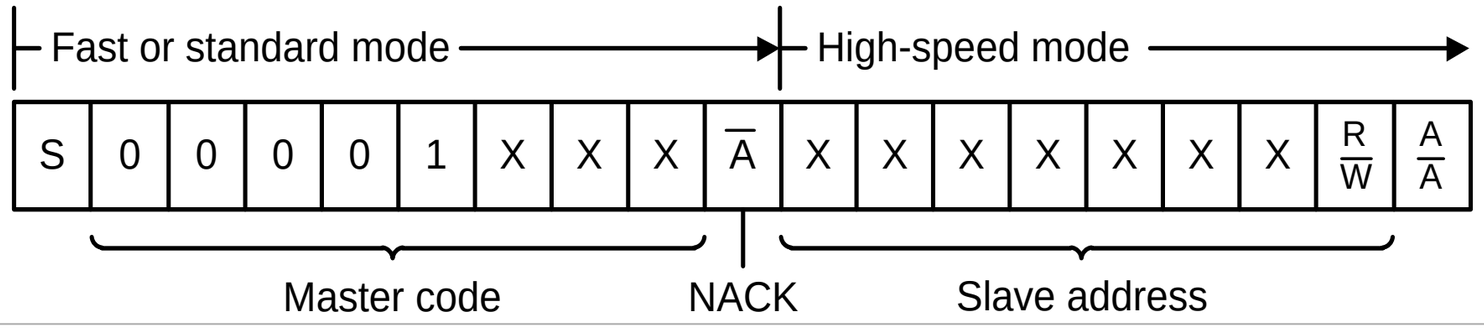
Reserved for future purposes
- Address 0x03

I2C Reserved Addresses – Hs controller Code

Target Address	R/$\overline{W}$ Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode controller code
111 11XX	1	Device ID
111 10XX	X	10-bit target addressing

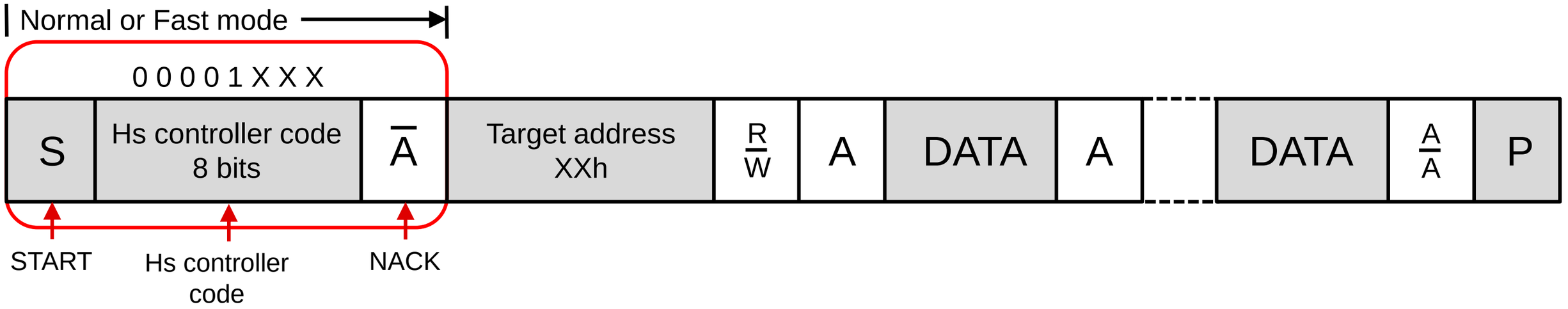
Hs-Mode controller code

- Code 0x04 to 0x07, not address
- Followed by mandatory NACK
- Last four bits used to identify the I2C controller
- Enables circuitry that allows for high-speed transfers
- A repeated start continues high-speed mode data transmission
- A STOP condition returns the I2C bus to fast or standard mode



I2C Reserved Addresses – Hs controller Code

High-Speed Mode

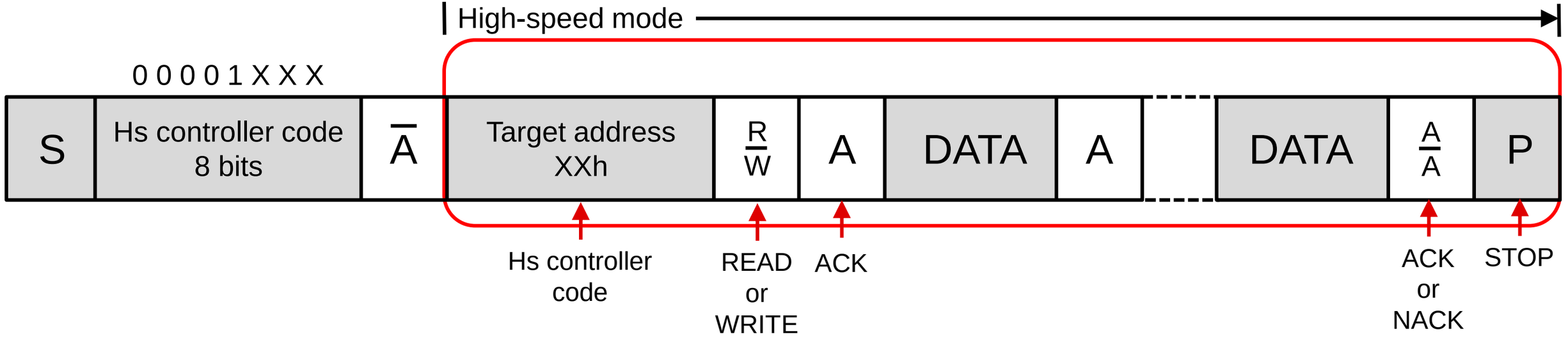


High-speed mode communication

- Controller sends a START condition
- Controller then sends the bits for the high-speed controller code (reserved address 04h to 07h)
- Since these bits are not an address, this communication is NACKed

I2C Reserved Addresses – Hs controller Code

High-Speed Mode



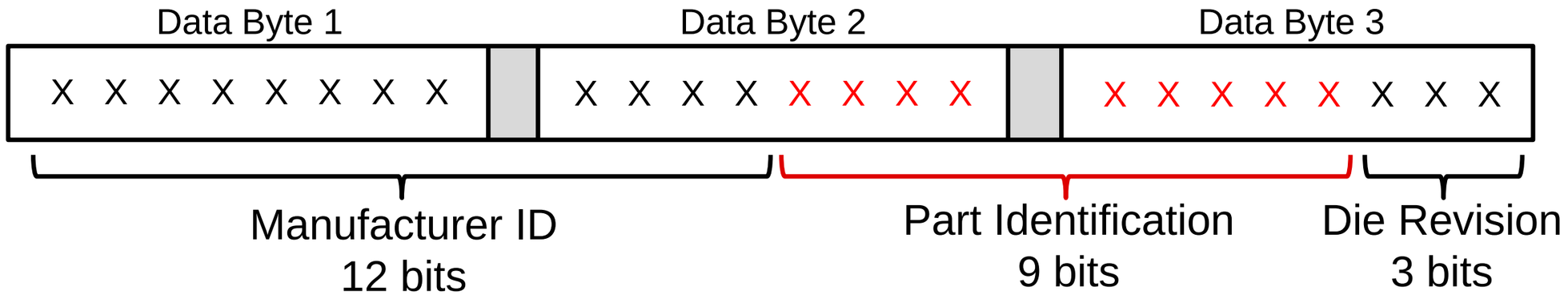
- Controller sends target address of high-speed mode device
- READ or WRITE bit is sent for communication
- Data is transmitted by controller or target device, with ACK for each data byte
- Target device stays in communication with controller until it receives a STOP
- A repeated START with a different target address will also stop communication, and start communication with the new address

I2C Reserved Addresses – Device ID

target Address	R/$\overline{W}$ Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode controller code
111 11XX	1	Device ID
111 10XX	X	10-bit target addressing

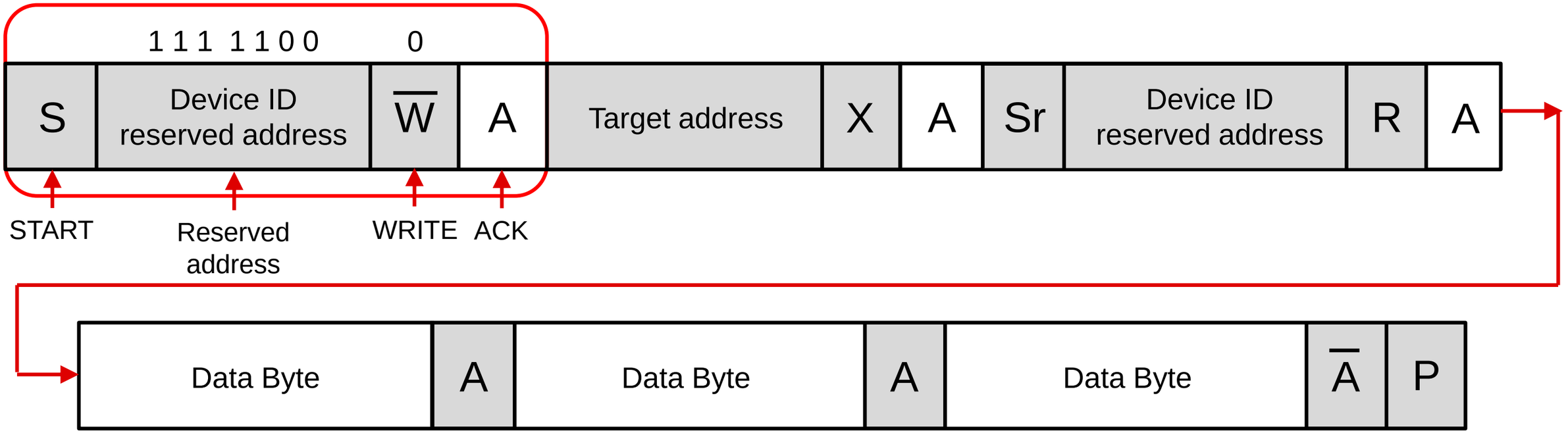
Device ID

- Addresses 0x7C – 0x7F
- Optional three byte (24 bit), read only word used for manufacturer information
- Fields:
  - 12 bits: Manufacturer ID
  - 9 bits: Part Identification
  - 3 bits: Die Revision



I2C Reserved Addresses – Device ID

READ the reserved address Device ID:

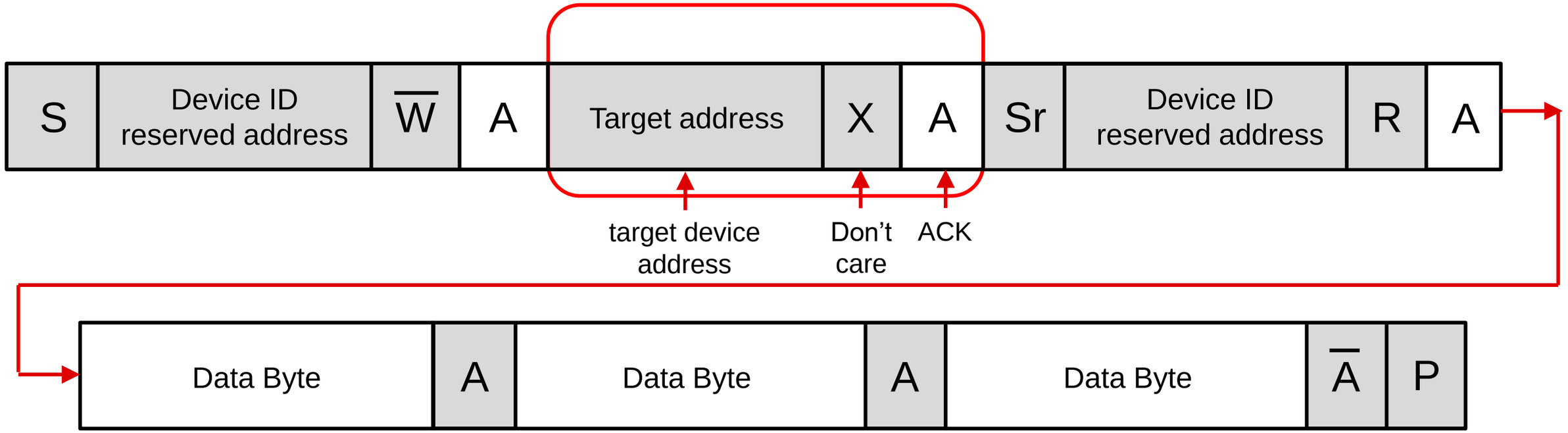


Using the reserved address:

- controller sends a START condition
- controller then sends the reserved address for Device ID
- The 8th bit is a 0 for write
- Any target device that recognizes Device ID sends ACK

I2C Reserved Addresses – Device ID

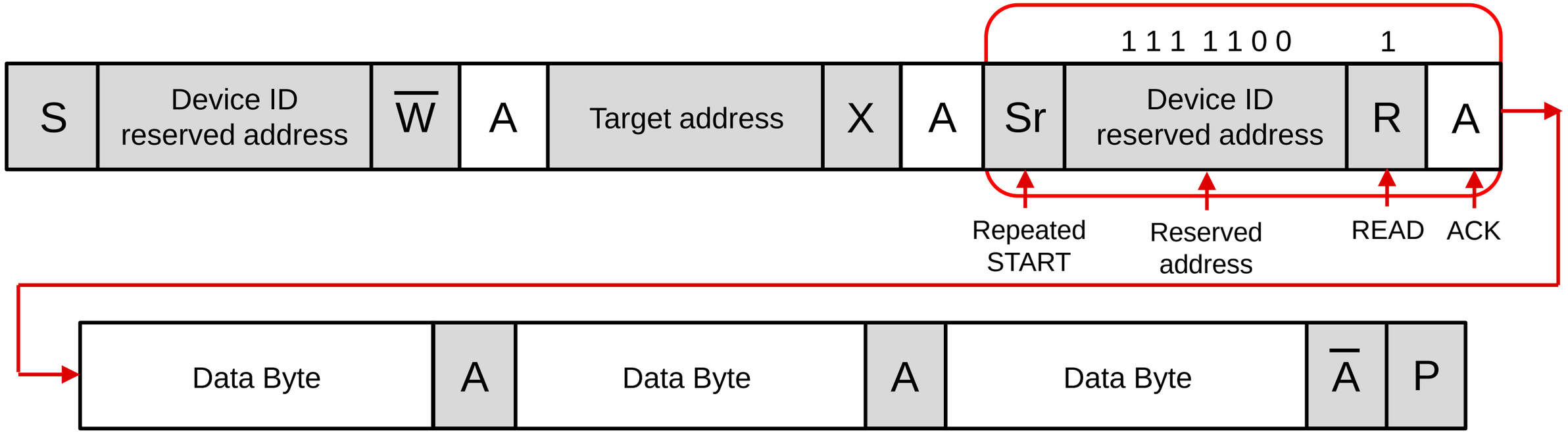
READ the reserved address Device ID:



- Controller then sends the address for the device it wants to ID
- The read write bit is then a don't care
- The target device responds with an ACK

I2C Reserved Addresses – Device ID

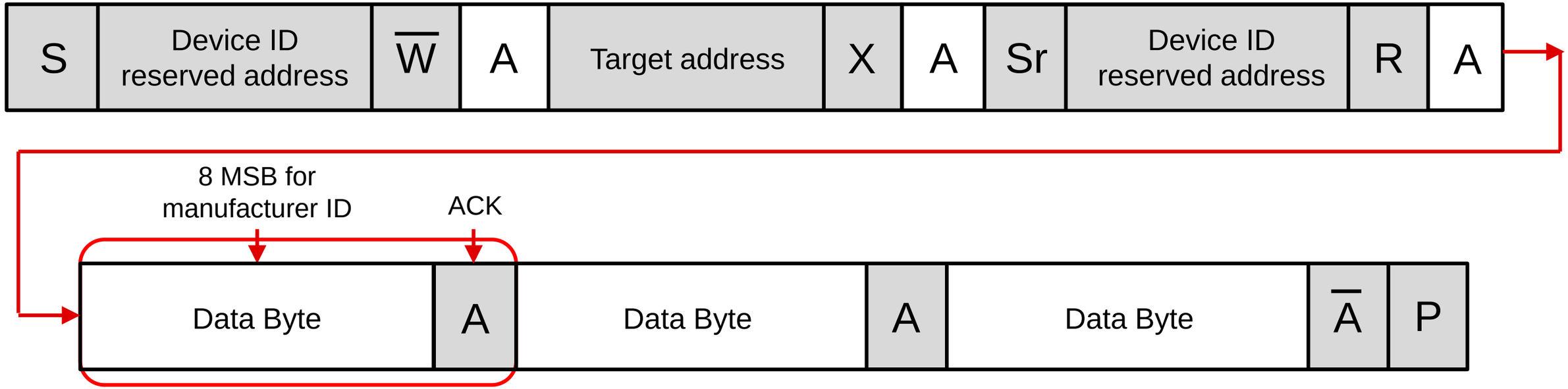
READ the reserved address Device ID:



- Controller then sends a repeated START
- Controller then sends the reserved address for Device ID
- The 8th bit is a 1 for read
- target device sends ACK

I2C Reserved Addresses – Device ID

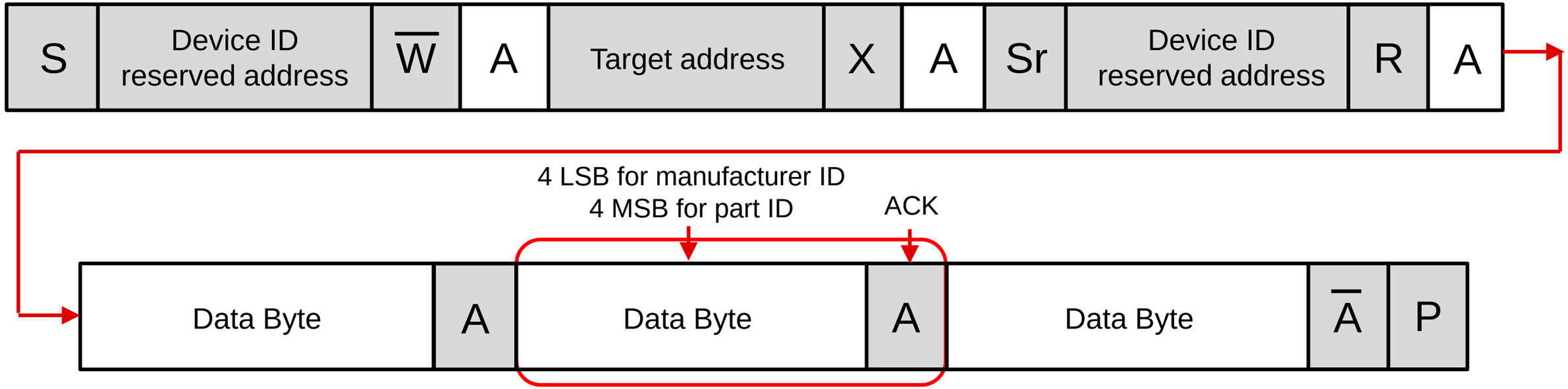
READ the reserved address Device ID:



- The target device then sends the Device ID
- The first data byte is the eight MSB for the 12-bit manufacturer ID
- Controller sends ACK that the data is received

I2C Reserved Addresses – Device ID

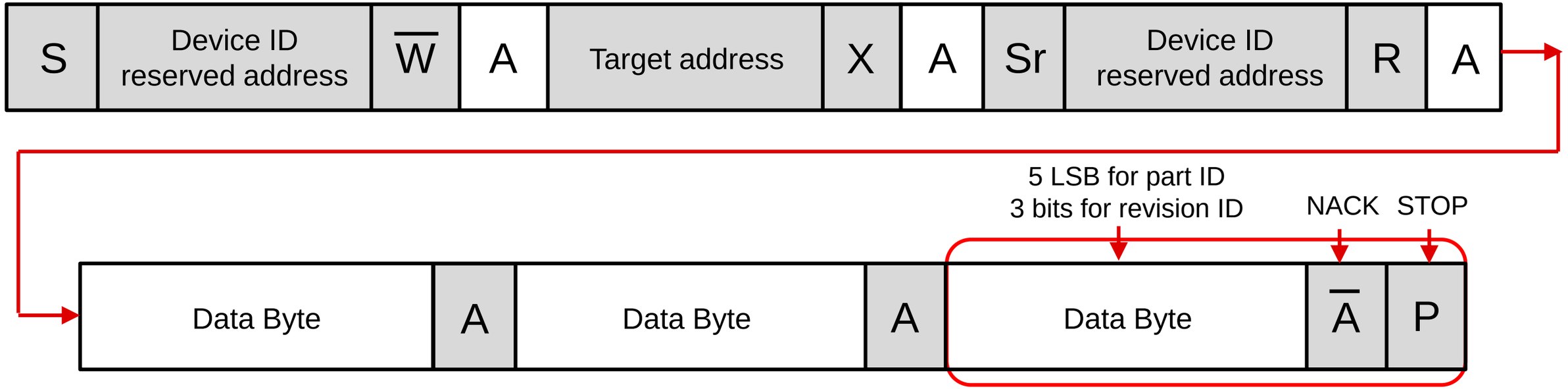
READ the reserved address Device ID:



- The second data byte starts with the four LSB for the 12-bit manufacturer ID
- The second data byte follows with the four MSB of the part ID
- Controller sends ACK that the data is received

I2C Reserved Addresses – Device ID

READ the reserved address Device ID:



- The third data byte starts with the five LSB for the 9-bit part ID
- The third data byte concludes with three bits for the revision ID
- Controller NACKs that the final byte and concludes the device ID data read with a STOP

I2C Reserved Addresses – 10-Bit target Addressing

target Address	R/W Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode controller code
111 11XX	1	Device ID
111 10XX	X	10-bit target addressing

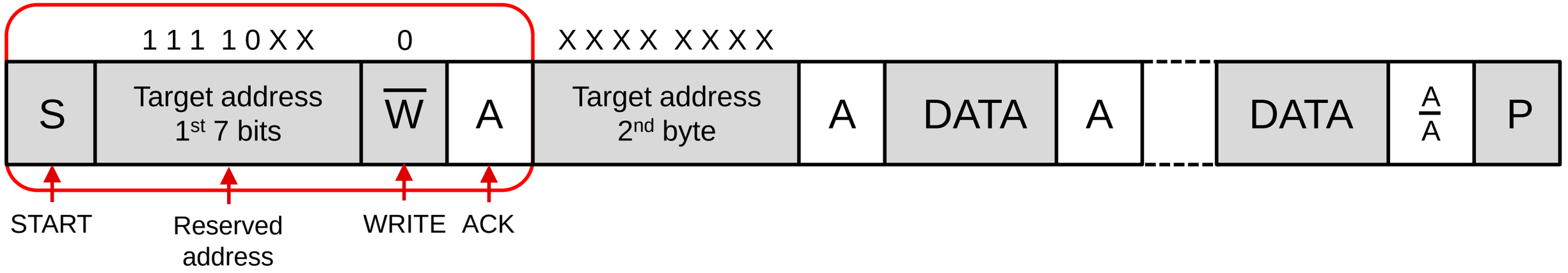
These two bits and the following byte make a total of 10 bits for addressing

10-Bit target Addressing

- Addresses 0x78 – 0x7B
- Expands I2C address to 10 bits with two bytes
- The 8th bit of the first byte still acts as the read/write
- The second byte completes the 10-bit address

I2C Reserved Addresses – 10-Bit target Addressing

WRITE with 10-bit address:

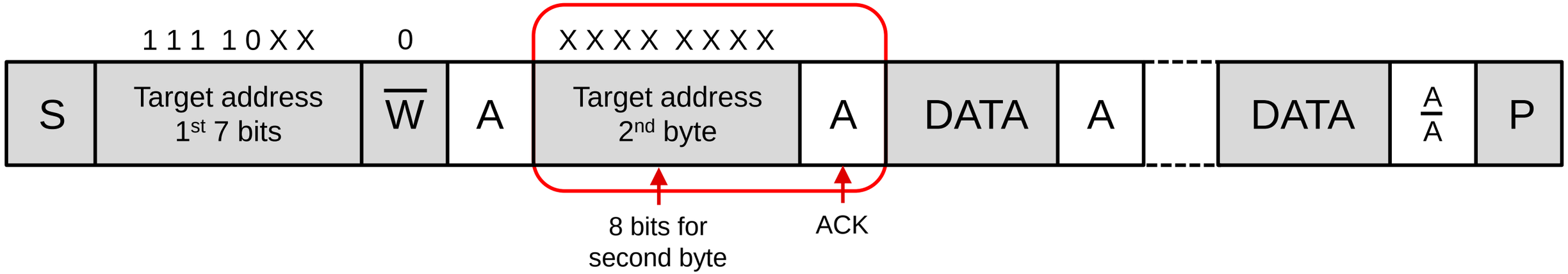


10-Bit target Addressing (WRITE):

- Controller sends a START condition
- Controller then send the 7 bits of the reserved address 78h to 7Bh
- The 8th bit of the first byte still acts as the read/write, here 0 for write
- Any 10-bit addressed target device that recognizes this address sends ACK

I2C Reserved Addresses – 10-Bit target Addressing

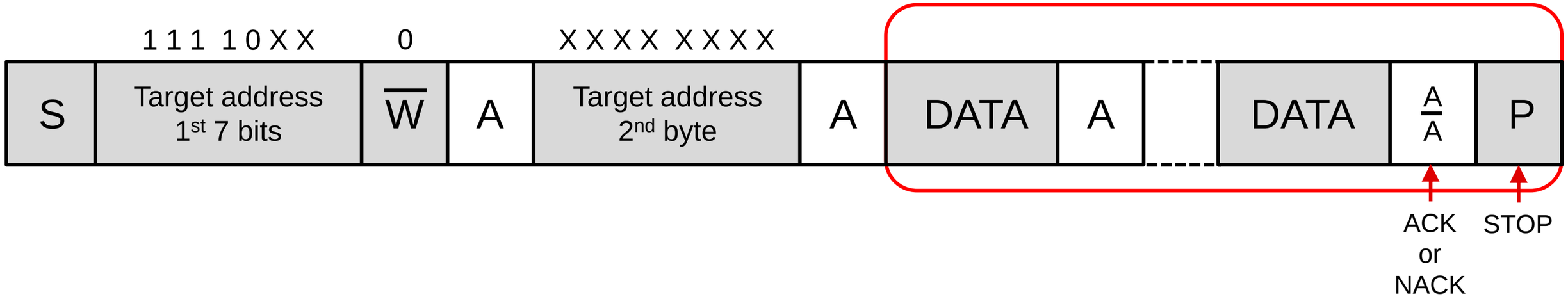
WRITE with 10-bit address:



- Controller then sends the second byte of the target address (8 bits)
- If correctly addressed, there should only be one device to ACK
- With the two bits of the reserved address and the eight bits of the second byte target address, this totals 10-bits of addressing.

I2C Reserved Addresses – 10-Bit target Addressing

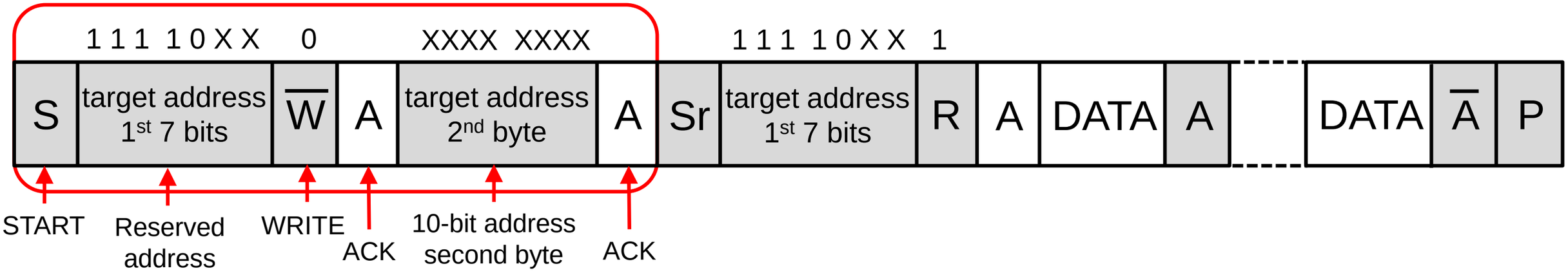
WRITE with 10-bit address:



- I2C communication continues normally with data byte transactions
- Target device stays in communication with controller until it receives a STOP
- A repeated START with a different target address will also stop communication, and start communication with the new address.

I2C Reserved Addresses – 10-Bit target Addressing

READ with 10-bit address:

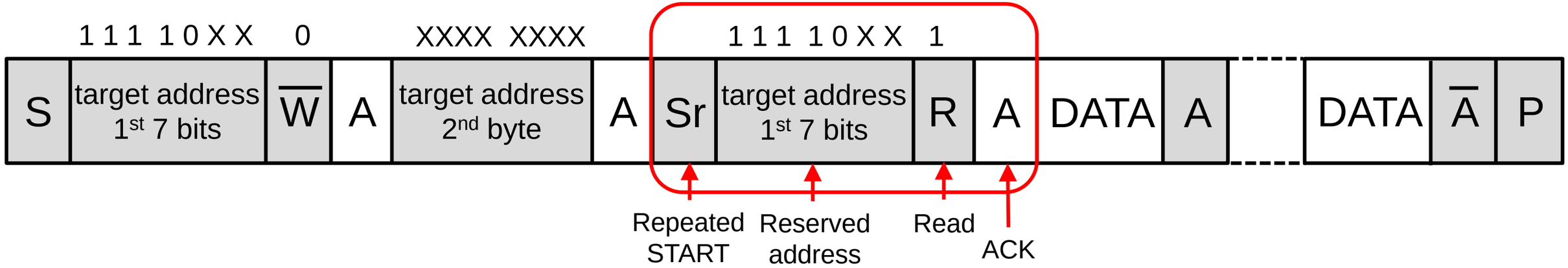


10-Bit target Addressing (READ):

- Starts exactly the same as a WRITE for 10-bit target addressing
- controller sends a START condition, then reserved address first 7 bits (78h to 7Bh)
- Then sends the WRITE (0) (same as a 10-bit write)
- Controller then sends target address 2nd byte for full 10-bit address

I2C Reserved Addresses – 10-Bit target Addressing

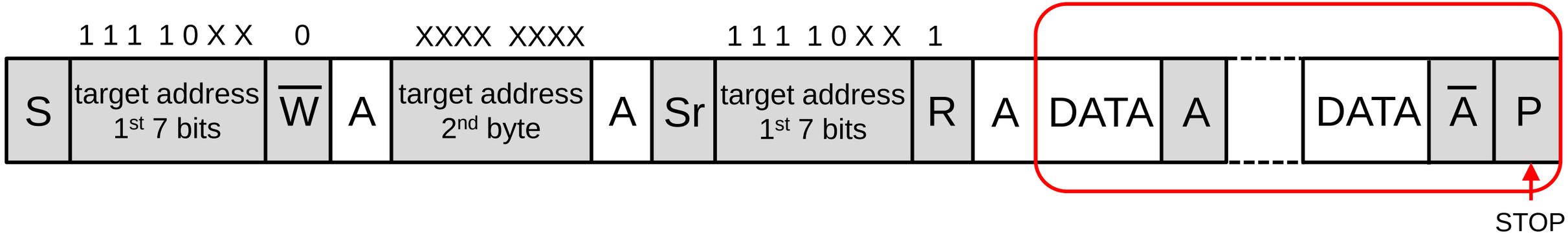
READ with 10-bit address:



- Controller then issues a repeated START and sends the same reserved address
- Controller then sends the READ bit (1)
- Device then ACKs the communication

I2C Reserved Addresses – 10-Bit target Addressing

READ with 10-bit address:



- I2C communication continues normally with data byte transactions
- Target device stays in communication with controller until it receives a STOP
- A repeated START with a different target address will also stop communication, and start communication with the new address.

Thanks for your time!
Please try the quiz.

Quiz: Basics of I2C: Reserved Addresses

1. Which of the following does not use a I2C reserved address?
   a. General call
   b. Device Acknowledge (ACK)
   c. High-speed controller code
   d. 10-bit addressing

Quiz: Basics of I2C: Reserved Addresses

1. Which of the following does not use a I2C reserved address?
   a. General call
   b. Device Acknowledge (ACK)
   c. High-speed controller code
   d. 10-bit addressing

Quiz: Basics of I2C: Reserved Addresses

2. How is I2C high-speed mode started?
   a. Write to the device's I2C address and the device's configuration byte
   b. Start by sending a high-speed mode controller code
   c. No special configuration is required for setting the device in high-speed mode
   d. Use the 10-bit address to set the high-speed mode

Quiz: Basics of I2C: Reserved Addresses

2. How is I2C high-speed mode started?
   a. Write to the device's I2C address and the device's configuration byte
   b. Start by sending a high-speed mode controller code
   c. No special configuration is required for setting the device in high-speed mode
   d. Use the 10-bit address to set the high-speed mode

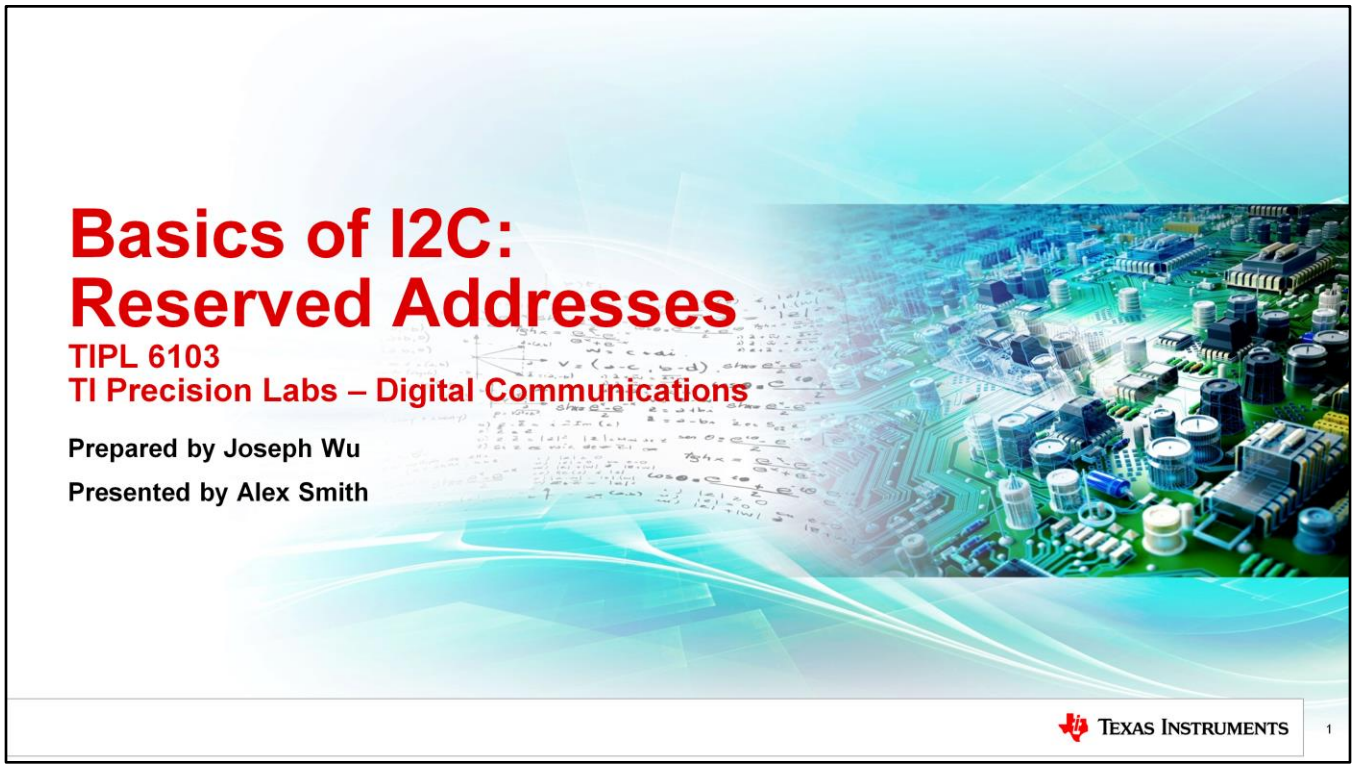
Quiz: Basics of I2C: Reserved Addresses

3. Which of the following is not identified by Device ID data
   a. Device function
   b. Manufacturer ID
   c. Part identification
   d. Die Revision

Quiz: Basics of I2C: Reserved Addresses

3. Which of the following is not identified by Device ID data
   a. Device function
   b. Manufacturer ID
   c. Part identification
   d. Die Revision

Thanks for your time!



Hello and welcome to our in-depth look at communications with precision data converters. In the I2C protocol, writing to and reading from the target device requires the use of an I2C address. The I2C address identifies which device the controller wants to communicate with. Typically, this address is written over a single byte, where the address itself is 7 bits, and an additional bit is used to indicate a read from or write to the device.

In other videos, we discuss the communication protocol and show how I2C addressing is done. However, not all addresses of the 7 bits can be used for target devices. In this video, we'll show that some addresses are reserved for other purposes and we'll show what functions these reserved addresses may be used for.

I2C Reserved Addresses

Target Address	R/$\overline{W}$ Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode controller code
111 11XX	1	Device ID
111 10XX	X	10-bit target addressing

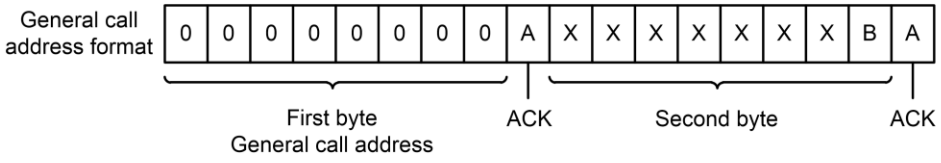
I2C has several sets of reserved addresses that are limited for use based on specific applications. Again, the standard I2C address is a 7-bit word. Theoretically, that would imply that there are 128 7-bit addresses available for devices on the I2C bus.

However, some of those 128 addresses are reserved. This table shows addresses that are reserved for I2C communications. The functions called with these reserved addresses are options for devices, and are not necessarily available in all I2C devices.

I2C Reserved Addresses – General Call

Target Address	R/$\overline{W}$ Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode controller code
111 11XX	1	Device ID
111 10XX	X	10-bit target addressing

General call address
- Addressed as 0x00 write
- Communicates to all devices
- Second byte contains command
- Can be used for several functions including RESET



Texas Instruments 3

The first reserved address is I2C address 0 and is the general call address. A write from the general call address is used to address all the devices connected to the I2C bus at the same time. Not all devices are designed to respond to the general call address. However, if the device does respond, then it may process a second byte and the following bytes after the general call.

If the LSB of the second byte is a zero, the second byte may be used to send commands to those devices receiving the general call. For example, one of the common commands sent through the general call is a reset. If the second byte is 06h, the controller device sends a general call reset to all devices on the I2C bus.

If the LSB of the second byte is a one, the controller is sending a hardware general call. In this case, the controller device may be a hardware controller that may not be programmed to transmit to a particular target address or it may not know what device it needs to send the command to. Instead, the second byte is used to send the master's own address to identify itself to all the devices on the system. This address can then be recognized by another processor device in the system, that can be used to direct information from the hardware controller.

I2C Reserved Addresses – START Byte

Target Address	R/$\overline{W}$ Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode controller code
111 11XX	1	Device ID
111 10XX	X	10-bit target addressing

START Byte

- Addressed as 0x00 read
- Used for devices that may be polling the SDA and SCL lines instead of using an integrated I2C controller
- Devices can poll at a slower rate until it detects a 0 on the bus.
- The device can then switch to higher polling rates to detect the I2C transaction
- Once the I2C transaction is completed, the device can resume slower polling

A read from I2C address 0 is the START byte. Not all microprocessors have a built-in, on-board I2C controller and it may not have an interrupt to detect communication on the bus. In that case, a microprocessor monitoring the I2C bus would need to repeatedly poll the SDA and SCL lines to ensure communications aren't missed. This requires the device to poll at a fast rate to detect the I2C address.

One option would be to use a START byte before the communication. When the controller uses the START byte with 7 zeros in the address, the target device can poll at a much slower rate, saving processing power.

After the target device detects that the SDA has sent a zero during the START byte, it can then switch to faster polling to detect the next I2C transmission for the address being sent. Once the I2C transfer has ended with a STOP condition, the target device can then resume polling at a slower rate, again saving processing power.

I2C Reserved Addresses – Other Protocols

Target Address	R/$\overline{W}$ Bit	Description
0000 000	0	General call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Reserved for different bus format
0000 011	X	Reserved for future purposes
0000 1XX	X	Hs-mode controller code
1111 1XX	1	Device ID
1111 0XX	X	10-bit target addressing

CBUS address
- Address 0x01
- Ignored by I2C bus devices
- CBUS no longer in use

Reserved for different bus format
- Address 0x02
- Allows for mixed protocols
- Only I2C devices than can work with different formats and protocols may respond to this address

Reserved for future purposes
- Address 0x03

The next three I2C addresses listed in the table are all reserved for different reasons.

Address 01 is reserved for the CBUS protocol so that CBUS devices could be placed on the I2C bus. However, the CBUS protocol is no longer used and this I2C address is ignored by most devices.

Address 02 is reserved for different bus formats. This is designed to allow communication between different protocols. Only I2C devices that work between different protocols can respond to this address.

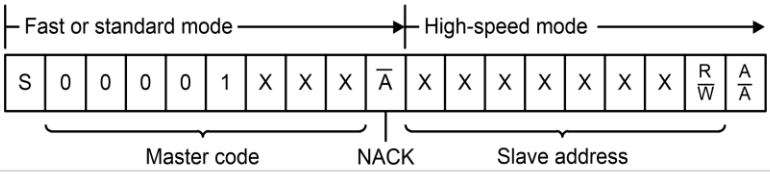
Address 03 is reserved for future purposes.

I2C Reserved Addresses – Hs controller Code

Target Address	R/$\overline{W}$ Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode controller code
111 11XX	1	Device ID
111 10XX	X	10-bit target addressing

Hs-Mode controller code

- Code 0x04 to 0x07, not address
- Followed by mandatory NACK
- Last four bits used to identify the I2C controller
- Enables circuitry that allows for high-speed transfers
- A repeated start continues high-speed mode data transmission
- A STOP condition returns the I2C bus to fast or standard mode



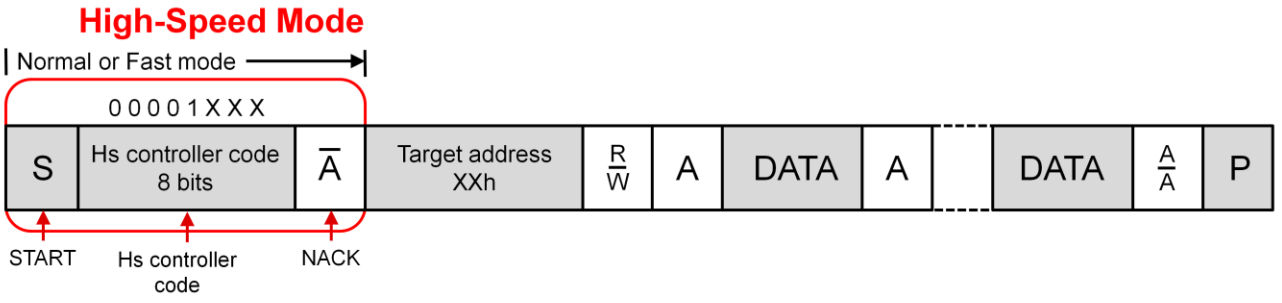
Texas Instruments

The next reserved addresses are for the High-speed controller code. These codes are from 04 to 07. The eighth bit normally used for read/write indication is used as part of the high-speed controller code. These high-speed controller codes are reserved 8-bit codes which are not used for target addressing or other purposes. Each high-speed controller has its own unique controller code and this allows for up to 8 high-speed masters on the I2C bus. The controller code for a high-speed mode controller device is software programmable and is chosen by the system designer.

Devices that support high-speed mode start operation in standard or fast mode. The controller code enables high-speed mode.
The high-speed controller code allows for arbitration between the high-speed masters and it indicates the start of a high-speed mode transfer. It enables internal current sources allowing the I2C communication bus to be faster than with just pull-up resistors.

When enabled, high-speed data transfer continues through the data transmission. A repeated start continues high-speed mode data transmission, while a STOP condition returns the I2C bus to fast or standard mode. This diagram shows the beginning of a high-speed mode transmission.

I2C Reserved Addresses – Hs controller Code



High-speed mode communication

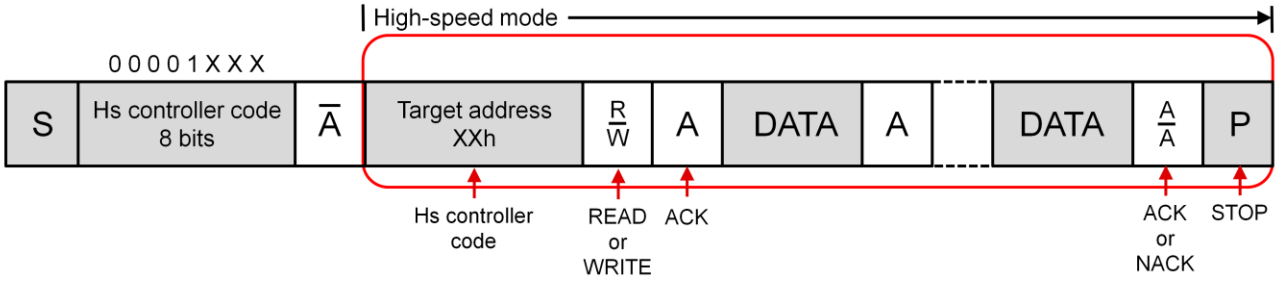
- Controller sends a START condition
- Controller then sends the bits for the high-speed controller code (reserved address 04h to 07h)
- Since these bits are not an address, this communication is NACKed

Here, we'll show how a controller writes the high-speed controller code to the device.

First, the device starts in standard or fast mode. The controller sends a START condition. Then the first byte is sent with the reserved address used for the high-speed controller code. Again, this isn't an address, and it uses the full 8 bits for the controller code. Because this value sent is not an address, there is no response from the device, and it is not acknowledged or NACKed. However, the controller code enables high-speed mode for all devices that are capable of high-speed mode. Even though the controller code is NACKed, the device's internal circuits for high-speed mode are enabled.

I2C Reserved Addresses – Hs controller Code

High-Speed Mode



- Controller sends target address of high-speed mode device
- READ or WRITE bit is sent for communication
- Data is transmitted by controller or target device, with ACK for each data byte
- Target device stays in communication with controller until it receives a STOP
- A repeated START with a different target address will also stop communication, and start communication with the new address

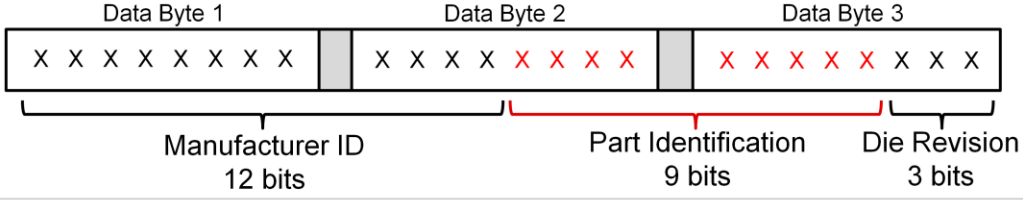
The controller then sends the target address of the high-speed mode device and follows with a READ or WRITE bit for communication.

Data is transmitted by the controller or target, with ACK for each data byte similar to the standard I2C communication. The target device continues communication until it receives a STOP condition or it receives a repeated start for a new target address.

I2C Reserved Addresses – Device ID

target Address	R/$\overline{W}$ Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode controller code
111 11XX	1	Device ID
111 10XX	X	10-bit target addressing

Device ID
- Addresses 0x7C – 0x7F
- Optional three byte (24 bit), read only word used for manufacturer information
- Fields:
  - 12 bits: Manufacturer ID
  - 9 bits: Part Identification
  - 3 bits: Die Revision



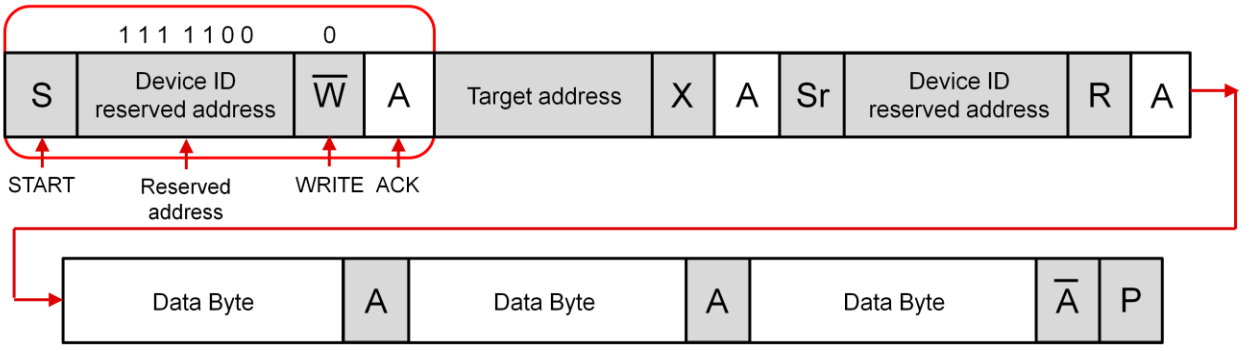
Texas Instruments 9

Addresses 7C to 7F are all reserved for Device ID. The controller starts by sending the Reserved Device ID address followed by a write bit. The controller then sends the target device address that it wants to identify. The controller then sends a Re-START condition followed by the Reserved Device ID address followed by a read bit.

The diagram shows the data format from the three bytes of Device ID. This data starts with 12 bits for the manufacturer ID, followed by 9 bits for the part identification, completed by 3 bits for the die revision.

I2C Reserved Addresses – Device ID

READ the reserved address Device ID:



Using the reserved address:

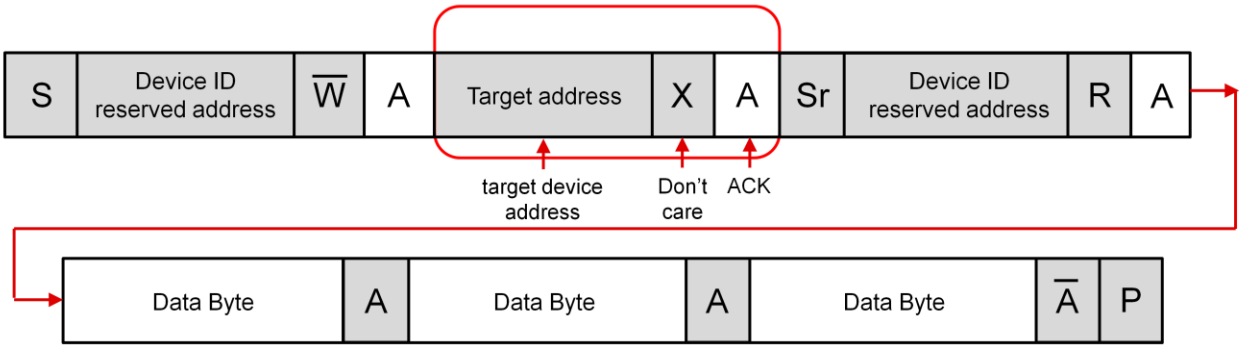
- controller sends a START condition
- controller then sends the reserved address for Device ID
- The 8th bit is a 0 for write
- Any target device that recognizes Device ID sends ACK

Here, we'll show how a controller reads information with the device ID address

First, the controller sends a START condition. The first byte is sent with the reserved address for Device ID and is followed by a 0 for a write. At this point there may be multiple slaves that may respond to Device ID, so multiple devices may ACK this address.

I2C Reserved Addresses – Device ID

READ the reserved address Device ID:

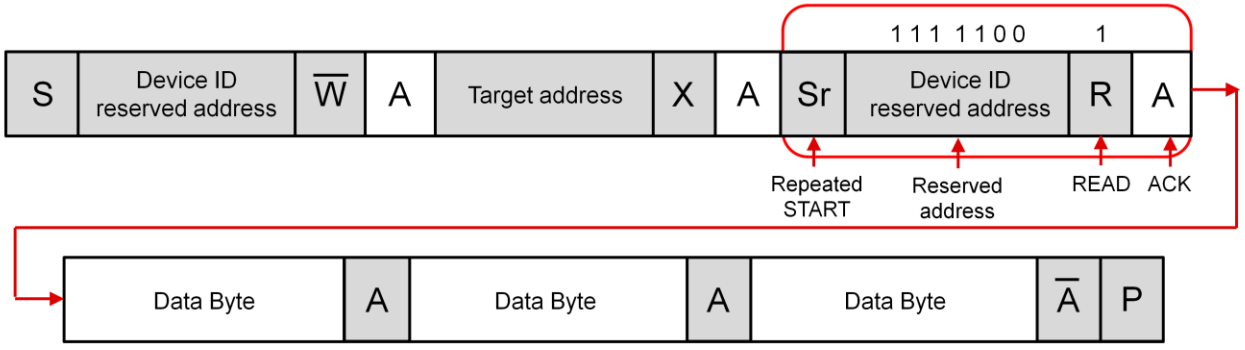


- Controller then sends the address for the device it wants to ID
- The read write bit is then a don't care
- The target device responds with an ACK

The controller then sends the address for the target device. The last bit of this target address byte is a "Don't Care" followed by an ACK. There should be only one device here that ACKs this address.

I2C Reserved Addresses – Device ID

READ the reserved address Device ID:



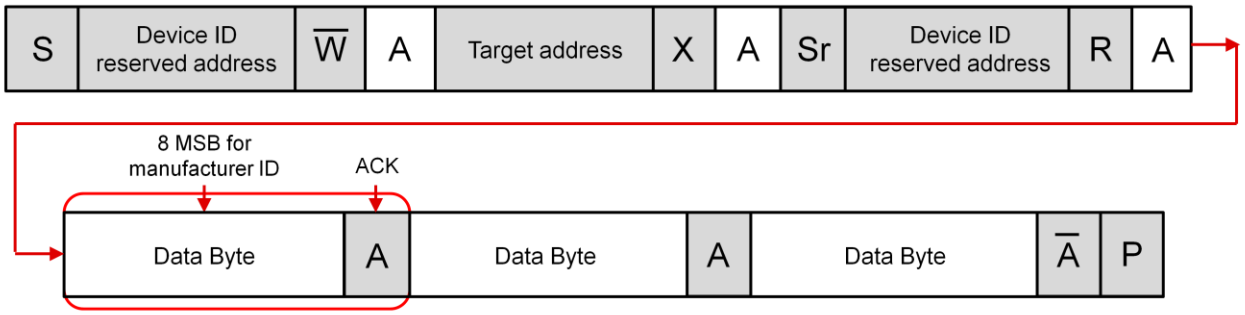
- Controller then sends a repeated START
- Controller then sends the reserved address for Device ID
- The 8th bit is a 1 for read
- target device sends ACK

The controller then sends a Re-START condition. After that, the controller sends the Reserved Device ID I2C-bus address followed by the Read bit. The target device ACKs this reserved address.

Note that the beginning of this third byte must be a Re-START. A STOP followed by a START condition, or a STOP with a re-START condition followed by access to a different target device resets the target device state machine and the Device ID read cannot be performed.

I2C Reserved Addresses – Device ID

READ the reserved address Device ID:

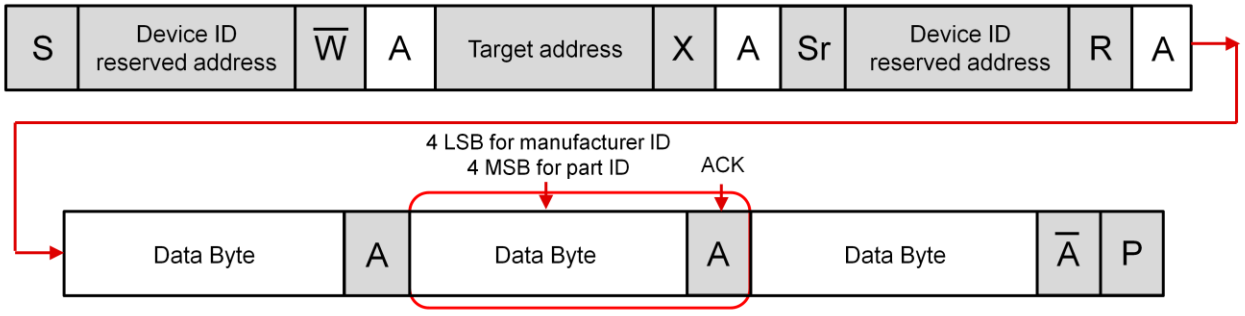


- The target device then sends the Device ID
- The first data byte is the eight MSB for the 12-bit manufacturer ID
- Controller sends ACK that the data is received

The target device then sends three bytes for device ID. The first byte gives the eight MSBs of the 12-bit manufacturer ID. The controller then ACKs the receipt of this byte.

I2C Reserved Addresses – Device ID

READ the reserved address Device ID:

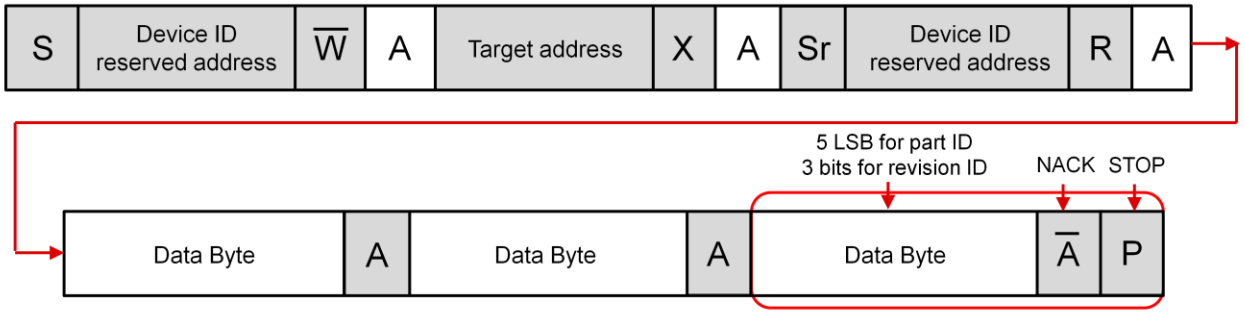


- The second data byte starts with the four LSB for the 12-bit manufacturer ID
- The second data byte follows with the four MSB of the part ID
- Controller sends ACK that the data is received

The target device then sends the second byte for Device ID. The four MSB of the byte are the 4 LSB of the manufacturer ID. The four LSB of the byte are the four MSB of the part identification. The controller ACKs the receipt of the second byte.

I2C Reserved Addresses – Device ID

READ the reserved address Device ID:



- The third data byte starts with the five LSB for the 9-bit part ID
- The third data byte concludes with three bits for the revision ID
- Controller NACKs that the final byte and concludes the device ID data read with a STOP

The target device finally sends the third byte for Device ID. The five MSB of this byte are the five LSB for the part identification. The three LSB of this byte are the three bits used for die revision. The controller NACKs the last byte and concludes the Device ID read with a STOP.

The reading of the Device ID can be stopped at anytime by sending a NACK. If the controller continues to ACK the bytes after the third byte, the target rolls back to the first byte and keeps sending the Device ID sequence until a NACK has been detected.

I2C Reserved Addresses – 10-Bit target Addressing

target Address	R/W Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode controller code
111 11XX	1	Device ID
111 10XX	X	10-bit target addressing

These two bits and the following byte make a total of 10 bits for addressing

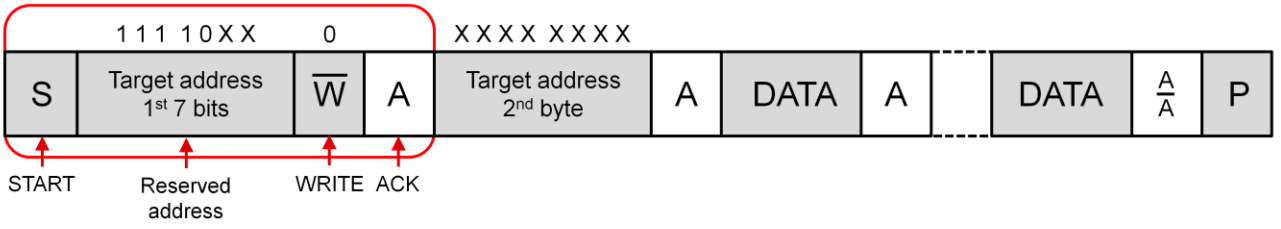
10-Bit target Addressing
- Addresses 0x78 – 0x7B
- Expands I2C address to 10 bits with two bytes
- The 8th bit of the first byte still acts as the read/write
- The second byte completes the 10-bit address

With the normal 7-bit I2C address and all of the reserved addresses, the number of possible I2C devices on a bus becomes limited.

To expand the number of devices, several reserved addresses can be used to expand the address to 10-bits. In the reserved address, the last two bits of 78h to 7Bh represent the first two bits used to expand the address space. A second full byte of eight bits is used to complete the 10-bit address. Communication with the 10-bit address is very similar to the 7-bit address communication.

I2C Reserved Addresses – 10-Bit target Addressing

WRITE with 10-bit address:



10-Bit target Addressing (WRITE):

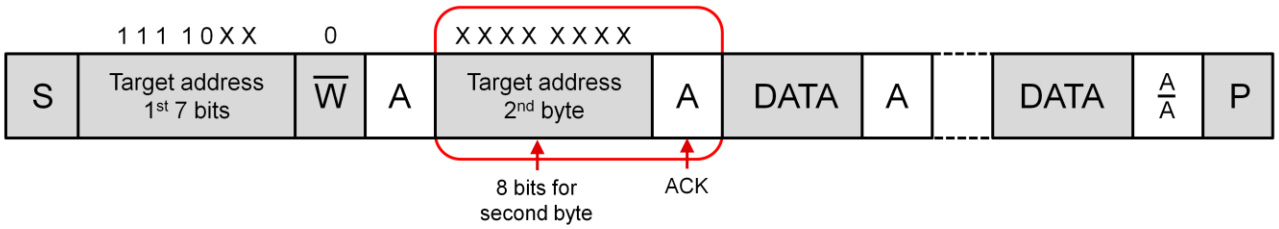
- Controller sends a START condition
- Controller then send the 7 bits of the reserved address 78h to 7Bh
- The 8th bit of the first byte still acts as the read/write, here 0 for write
- Any 10-bit addressed target device that recognizes this address sends ACK

Here, we'll show how a controller writes to a device with a 10-bit address.

First, the controller sends a START condition. Then the first byte is sent with the reserved address for 10-bit I2C addressing and is followed by a 0 for a write. At this point there may be multiple slaves that have the same first address byte for 10-bit addressing, so multiple devices may ACK this address.

I2C Reserved Addresses – 10-Bit target Addressing

WRITE with 10-bit address:



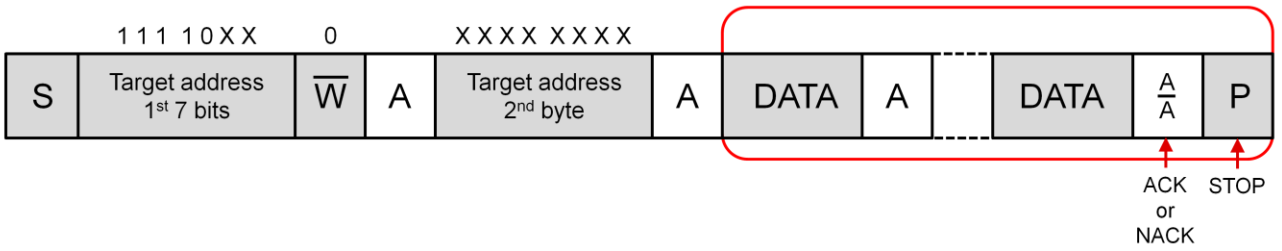
- Controller then sends the second byte of the target address (8 bits)
- If correctly addressed, there should only be one device to ACK
- With the two bits of the reserved address and the eight bits of the second byte target address, this totals 10-bits of addressing.

Then the controller device sends the second byte of the target address. With the second byte, there should be only one device with the unique 10-bit address. This is the only device that should ACK the communication.

With the two bits of the reserved address and the eight bits of the second byte target address, this totals 10-bits of addressing.

I2C Reserved Addresses – 10-Bit target Addressing

WRITE with 10-bit address:



- I2C communication continues normally with data byte transactions
- Target device stays in communication with controller until it receives a STOP
- A repeated START with a different target address will also stop communication, and start communication with the new address.

This target stays in communication with the controller until it receives a STOP condition or the controller sends a repeated START condition to communicate with a different target address.

I2C Reserved Addresses – 10-Bit target Addressing

READ with 10-bit address:

10-Bit target Addressing (READ):

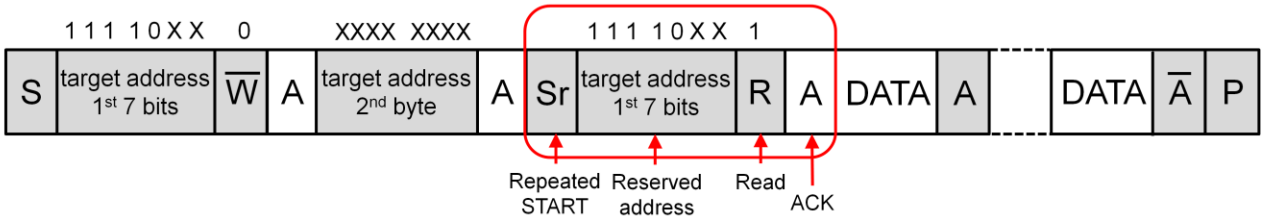
- Starts exactly the same as a WRITE for 10-bit target addressing
- controller sends a START condition, then reserved address first 7 bits (78h to 7Bh)
- Then sends the WRITE (0) (same as a 10-bit write)
- Controller then sends target address 2nd byte for full 10-bit address

Here, we'll show how a controller reads from a device with a 10-bit address.

Reading from a 10-bit addressed device is similar but with added steps. First there is a START followed by the reserved address, followed by a 0 for a write bit. This is followed by an ACK from all devices that use the reserved address. This is then followed by the target address second byte and then an ACK from the addressed device. Until this point, the communication is exactly the same as a 10-bit address write.

I2C Reserved Addresses – 10-Bit target Addressing

READ with 10-bit address:



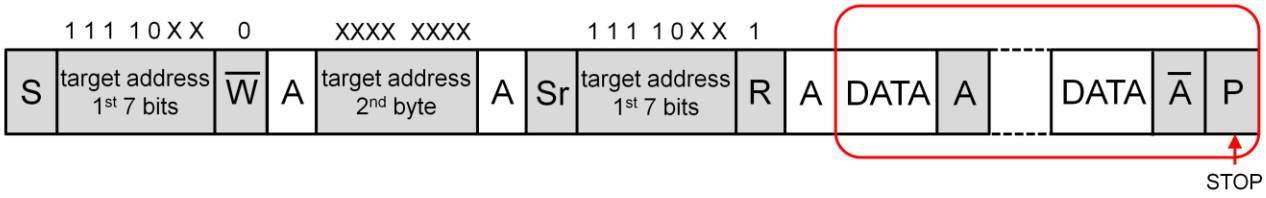
- Controller then issues a repeated START and sends the same reserved address
- Controller then sends the READ bit (1)
- Device then ACKs the communication

To read from this device, the controller then sends a repeated START. This is followed by the reserved address that was just used. Then the read bit is sent followed by an ACK.

Because the read bit is sent (and not the write bit), the device that just ACKed this communication understands that this is a read. Other devices with the same reserved address do not respond. The addressed device then ACKs this repeat of the reserved target address.

I2C Reserved Addresses – 10-Bit target Addressing

READ with 10-bit address:



- I2C communication continues normally with data byte transactions
- Target device stays in communication with controller until it receives a STOP
- A repeated START with a different target address will also stop communication, and start communication with the new address.

After the addressed device sends the ACK, data is transmitted by the device, and after each byte, the controller ACKs the data. This data transmission continues until the controller sends a STOP condition or a repeated START followed by a different target address.

That concludes this video – thank you for watching! Please try the quiz to check your understanding of this video's content.

Quiz: Basics of I2C: Reserved Addresses

1. Which of the following does not use a I2C reserved address?
   a. General call
   b. Device Acknowledge (ACK)
   c. High-speed controller code
   d. 10-bit addressing

Quiz: Basics of I2C: Reserved Addresses

1. Which of the following does not use a I2C reserved address?
   a. General call
   b. Device Acknowledge (ACK)
   c. High-speed controller code
   d. 10-bit addressing

Quiz: Basics of I2C: Reserved Addresses

2. How is I2C high-speed mode started?
   a. Write to the device's I2C address and the device's configuration byte
   b. Start by sending a high-speed mode controller code
   c. No special configuration is required for setting the device in high-speed mode
   d. Use the 10-bit address to set the high-speed mode

Quiz: Basics of I2C: Reserved Addresses

2. How is I2C high-speed mode started?
   a. Write to the device's I2C address and the device's configuration byte
   b. Start by sending a high-speed mode controller code
   c. No special configuration is required for setting the device in high-speed mode
   d. Use the 10-bit address to set the high-speed mode

Quiz: Basics of I2C: Reserved Addresses

3. Which of the following is not identified by Device ID data
   a. Device function
   b. Manufacturer ID
   c. Part identification
   d. Die Revision

Quiz: Basics of I2C: Reserved Addresses

3. Which of the following is not identified by Device ID data
   a. Device function
   b. Manufacturer ID
   c. Part identification
   d. Die Revision

Thanks for your ti