EEPROM Design Guide

- **Write Protect CAT24WCxxx I^2C™ Serial EEPROMs.**
  - Allows the user to protect against inadvertent write operations.

    WP = V_{CC} : *Write Protected*

    Device select and address bytes are Acknowledged

    Data Bytes are not Acknowledged

    WP = V_{SS} or allowed to Float: *Unprotected*

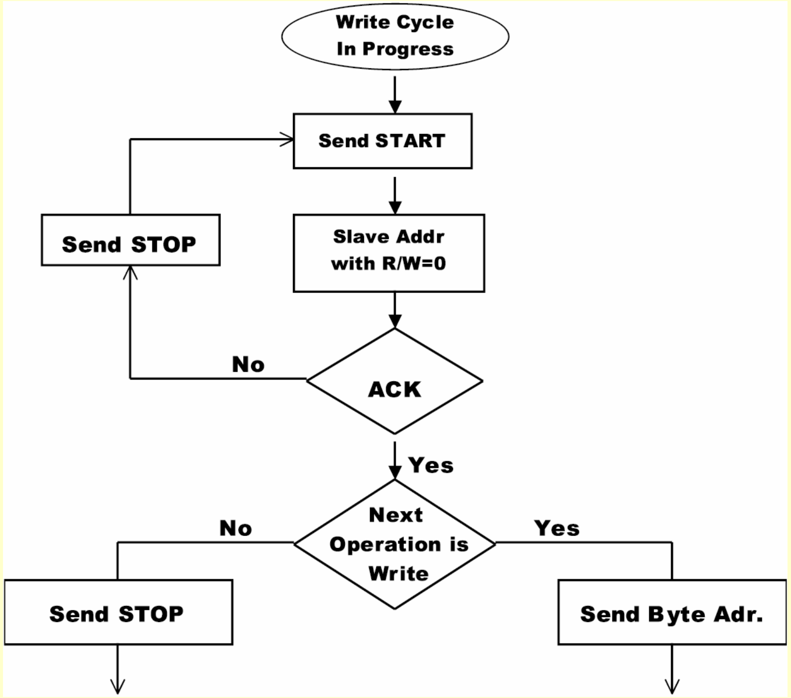
  - WP is sampled at the end of the acknowledge pulse after the last address byte. The WP signal should be in the desired state after sending the last bit of the address and should remain unchanged until after sending first bit of data.

EEPROM Design Guide

- **For CAT24WCxxx Devices, What Should the Pull-up Resistor Values be on the SDA and SCL Lines?**
  - The value of pull-up resistor, R_P, depends on: supply voltage, bus capacitance (C_b), output driver sink current and input current of connected devices.
    - SDA line: R_P = 1.5 - 2.0kΩ - for fast operation / high C_b
    - SDA line: R_P = 2.0 - 16kΩ - for low power consumption / low C_b
  - The SCL line requires a pull-up resistor only if it is driven by an open-collector output.

EEPROM Design Guide

- **Minimize System Delays by Acknowledge Polling CAT24WCxxx Devices.**
  - During an internal write cycle, CAT24WCxxx devices do not respond to the slave address.
    - The write cycle time is the time from a valid stop after a write sequence to the end of the internal program/erase cycle.
    - Typical write time << t_{WR} max (Given in AC specifications)
    - For minimum system delay, an ACK polling sequence should be issued by the Master (See figure).



© 2001, Catalyst Semiconductor, Inc.
www.catalyst-semiconductor.com

EEPROM Design Guide

- **Initializing CAT24WCxxx I^{2}C™ Serial EEPROMs after an Interruption in the Transmission Protocol.**
  - The “START” condition will initiate the CAT24WCxxx and prepare it for the a new command.
  - The following sequence is recommended:

    1) Send a clock pulse on SCL (keep SDA HIGH from the master side)

    2) Test if SDA is HIGH (during SCL HIGH)

    3) If SDA is not HIGH, repeat from point 1) for up to 9 clock cycles

    4) If SDA is HIGH, generate a “START” condition

EEPROM Design Guide

- **Preventing Accidental Writing of CAT93Cxx MicrowireÔ Serial EEPROMs.**
  - CAT93CXX devices power-up in the Erase/Write Disable State.
  - Programming instructions must be preceded by an Erase/Write Enable (EWEN) instruction.
  - Protect against accidental writes by sending Erase/Write Disable (EWDS) instruction after any write to the device.
  - Avoid uncontrolled inputs during power up/down and microcontroller start-up time by using a pull-down resistor on the CAT93Cxx inputs.

EEPROM Design Guide

- **For MicrowireÔ Devices is it Necessary to Toggle CS to Initiate a Memory Write?**
  - For CAT93Cxx devices, CS toggling is necessary to initiate a memory write.
  - After receiving a Write Command, Address and Data, the CS pin must be immediately deselected for t_{CSMIN} after the last bit, D0.
  - The failing edge of CS starts the internal write cycle, measured by t_{EW} delay.

EEPROM Design Guide

- **Selecting CAT93Cxx Serial EEPROM Memory Organization.**
  - Internal memory organization of the CAT93Cxx can be selected using the ORG input:
    - ORG = V_{CC} or NC: x 16 Memory Organization
    - ORG = GND: x 8 Memory Organization
  - **Write and Read operations must be performed with the same memory organization selected.**

EEPROM Design Guide

- **Reducing CAT93Cxx Power Consumption.**

  - Current consumption is increased by several microamps (10µA max) when the ORG pin is connected to GND (x 8 organization).
  - To save power in the "x 8" mode, the ORG input can be switch to V_{CC} when the EEPROM is not accessed.
  - The ORG signal can be changed at the beginning of an instruction.
    - For **Read** instructions, ORG must be kept in the same state after the last clock of the instruction.
    - For **Write** instructions, ORG must be in the same state until the "Ready" status on the DO line is detected (or t_{EW} max after CS goes LOW at the end of the write instruction).

Note: ORG should be in the same state for write and read operations.

EEPROM Design Guide

- **What Timing Should be Used for the CAT93C86 Program Enable, PE, Signal?**
  - Write, Erase, Write All and Erase All instructions require PE="1".
  - PE must be kept at a "1" during an internal write cycle for a minimum of t_{EW} after CS toggles LOW.

EEPROM Design Guide

- **CAT93Cxx Serial EEPROM Power Up / Down Sequence Recommendations.**
  - **Power-up** sequence:
    - All the input signals to the CAT93Cxx should be at 0V.
    - Power-on V_{CC} with a ramp faster than 1V/10ms.
    - Wait at least 1 ms after V_{CC} has reached the nominal value before initiating the desired operation.
  - **Power-down** sequence:
    - All the inputs should be inactive in “0” state.
    - Power down with a ramp faster than 1V/10ms.

EEPROM Design Guide

- **For CAT25Cxxx SPI EEPROMs, Should CS be Toggled to Send New Commands?**
  - A HIGH to LOW transition on CS is required prior to any sequence being initiated.
  - After a valid write sequence, a LOW to HIGH transition on CS is necessary to initiate the internal write cycle.

- **What is the Difference Between SPI Mode (0,0) and (1,1)?**
  - Four possible data-clock timing relationships can be chosen by the control bits CPOL (Clock Polarity) and CPHA (Clock Phase).
  - The most common SPI modes supported by SPI Serial EEPROMs are:
    - (CPOL, CPHA) = (0,0)
    - (CPOL, CPHA) = (1,1).
    - For these two modes, input data is latched by the positive clock (SCK) edge, and output data is available on negative clock transitions.
  - The difference between SPI modes (0,0) and (1,1) is the stand-by polarity, when there is no data transfer:
    - SCK remains at “0” for (CPOL, CPHA) = (0,0)
    - SCK remains at “1” for (CPOL, CPHA) = (1,1)

EEPROM Design Guide

- **Write Sequence for SPI™ Serial EEPROM Devices.**
  - Writing to Status Register:
    - Send Write Enable Instruction (WREN), Toggle CS, Send Write Status Register (WRSR).
  - Writing to the Memory:
    - Send Write Enable Instruction (WREN), Toggle CS, Send Write Instruction.
  - ***CAT25Cxxx SPI devices automatically return to the write disable state at the end of a write cycle.***

EEPROM Design Guide

- **Write Time of Block Protect Bits for CAT25Cxxx SPI Serial EEPROMs.**
  - Write time for the Status Register bits, BP0 and BP1, is the same as for a write operation into the memory, t_{WC}.
  - BP0, BP1 are nonvolatile bits.

EEPROM Design Guide

- **CAT25Cxxx Programmable Hardware Write Protect Feature.**
  - – Hardware write protection is enabled for:
    - **WP = “0”** and Status Register bit **WPEN = “1”**
  - – Prevents the Status Register (BP0, BP1, WPEN) and protected memory blocks from being written.

EEPROM Design Guide

- **Power-Up and Power-Down in CAT25xxx SPI EEPROM devices**
  - **While the supply ramps up from 0 V to the final level, the CAT25xxx state machines are reset by an internal Power- On Reset (POR) pulse.**
    - The POR pulse starts when the supply reaches approx. 0.8 V and is timed out on chip.
    - The recommended wait time for applying commands is specified (conservatively) from the moment when the supply reaches nominal level.
  - **When powering down, the supply should be taken all the way to 0 V.**
    - Powering down to brownout levels (0.2 – 0.8V) should be avoided, as the internal POR may not be properly generated when the supply recovers.
    - The 1st write operation following brownout may not execute correctly.
  - **If brownout is unavoidable, then rewriting the Status Register contents will reset the CAT25xxx.**

EEPROM Design Guide

- **What is the Byte Load Cycle Time for Parallel EEPROMs?**
  - Byte Load Cycle time, t_{BLC}, is the interval during which the data byte will be latched in the page write mode.
  - If the next byte is not loaded within t_{BLC} max, the internal write cycle will start, and any additional attempts to load data will be ignored.
    - For Catalyst Parallel EEPROMs, t_{BLC} max = 100 µS

EEPROM Design Guide

- **Hardware Data Protection Feature for Catalyst Parallel EEPROMs**

  .

  – Hardware data protection is available through the following features:

    1. Voltage Level Sense Detector
       – Write Protect when V_{CC} < 3.5V min

    2. Power On Delay Timer
       – 5 to 10 ms delay before a write, after V_{CC} = 3.5V min

    3. Three-Line Write Control
       – Write is enabled only for: OE=1 and WE=0 and CE=0

    4. Noise Filter on WE and CE
       – Write cycles will not be initiated if pulses on WE and CE are less than 20 ns

EEPROM Design Guide

- **End of Write Operation Detection for Parallel EEPROM.**
  - Catalyst Parallel EEPROMs feature end of write cycle detection using:
    - Data Polling:
      - During the internal write cycle in progress, reading the last byte written will output the complement data on I/O7. Upon completion of the write operation, a read cycle will output the true data on I/O7.
    - Toggle Bit:
      - I/O6 toggles between “1” and “0” when a read data is performed d uring a write cycle in progress. Once the write is complete, I/O6 stops toggling.
    - RDY/BUSY Output:
      - This open drain output is active LOW during a write cycle and becomes inactive when the internal write operation completes (available with some products).

EEPROM Design Guide

- **Software Data Protection of Parallel EEPROMs.**
  - Software Data Protection (SDP) prevents data from being written to memory, unless an unlock sequence precedes the loading of the data byte.
    - To **enable** SDP, the user must send a specific write sequence.
      - Three specific data bytes are sent to three specific memory locations.
    - To **disable** SDP, the user must write specific data bytes into six specific locations.
  - SDP is non-volatile and is not changed at power up / down.
  - Catalyst EEPROM devices are shipped in the unprotected state.
    - It is highly recommended users enable the SDP feature. This increases system reliability by giving extra protection against data corruption during the power up/down sequence and noise pulses on the CE/WE lines.

EEPROM Design Guide

- **Possible Failures in EEPROMs.**
  - Loss of Data
    - Bit errors – Random failures during useful life of product
    - Due to “Charge gain”/ “Charge loss”
    - Resulting from data retention degradation after program/erase cycling
  - Read Failures
    - Degradation of read access time
    - Incorrect sensing of data (noise)
    - Improper read operation related to specifications (timing, voltage levels)
  - Write Failure
    - Loss of memory transistor functionality to program or erase data
    - Improper write operation related to specifications (timing, voltage levels)
  - Data Corruption Failures Caused by Inadvertent Write
    - During Power-Up/Down
    - Noise spikes on control lines