Bus Basics - Classic

Conceptual Level Block Diagram
Two Module Model

Master

Slave

Control Lines

Address Lines

Data Lines

Classic Write Sequence
Classic Read Sequence

![Classic Read Sequence Diagram](image)

Two & Three State Logic

![Two & Three State Logic Diagram](image)

**Figure 1.** A 2-input TTL NAND Gate with a Totem Pole Output Stage
Set Up, Hold Times

Bus Transactions - Clocked
Basic Concept: Master to Slave with Common Clock

Clock Line

- Master Control
- Slave Control
- Slave Control

Master Begins Process:

- Arbitration to determine ownership
- Initiate activity to communicate with slave
  - Assert address to identify target
  - If write, also assert data
  - Assert READ line appropriately (read or write)
  - Assert REQ line
- Activity to occur in single cycle, before active edge of clock
Slave Reaction to Master Activity

- On active edge of clock: accept address, direction (read line), REQ line
- During cycle, determine how to respond
  - Not our address: do nothing
  - Is our address: do appropriate work and then assert ACK

Master Write Activity
(after Arbitration)

- Place address on address lines
- Place data on data lines
- Place ‘0’ on READ line
- Assert REQ
- Wait for assertion of ACK
Slave Activity on Receiving Write Request

- Compare received address with own address
- If address doesn’t match: do nothing
- If address matches, determine how to respond
  - Place data value in appropriate place
  - Assert ACK
- Wait for release of REQ

Master Write Activity (Termination)

- When ACK detected, release bus
  - De-assert (tri-state) address bus
  - De-assert (tri-state) data bus
  - De-assert (tri-state) read line, REQ
- Wait for release of ACK
  - When release of ACK detected, system can move on to next transaction
Slave Activity – Terminating Write Transfer

- When release of REQ detected, terminate slave write activity
  - Release (tri-state) ACK
- System is now ready for next transaction

Master Read Activity
(After Arbitration)

- Place address on address bus
- Make sure data bus not asserted by master
- Place ‘1’ on READ line
- Assert REQ
- Wait for assertion of ACK
Slave Activity on Receiving Read Request

- Compare received address with own address
- If address doesn’t match: do nothing
- If address matches, get the information and place it on the data bus
  - If register, data is available so send to bus
  - If not register, go get it, then send to bus
  - Assert ACK when data ready
- Wait for release of REQ

Master Read Activity (Termination)

- When ACK detected, accept data
  - Place data in intended target
  - De-assert (tri-state) address bus
  - De-assert (tri-state) read line, REQ
- Wait for release of ACK
  - When release of ACK detected, system can move on to next transaction
Slave Activity – Terminating Read Transfer

- When release of REQ detected, terminate slave read activity
  - Release (tri-state) ACK
  - Release (tri-state) data bus
- System is now ready for next transaction

Activity of Master (after Arbitration)
Activity of Slave

Diagram:

- **S0**: Initial state
  - Transition to **S1** via **ADROK**
  - Transition to **S4** via **ADROK**

- **S1**: Accept Data to R
  - Transition to **S2** via **A(ACK)**

- **S2**: Transition to **S3** via **REQ**

- **S3**: Transition to **S0** via **R(ACK)**

- **S4**: Transition to **S5** via **A(DATA) A(ACK)**

- **S5**: Transition to **S0** via **R(ACK) R(DATA)**

Additional Details:

- **A(ACK)**: Accept Acknowledgment
- **R(ACK)**: Request Acknowledgment
- **DATA**: Data Signal
- **ADROK**: Accept Data Request OK

Diagram depicts the flow of states and transitions in the activity of a slave device or system.
Inverter circuit with open-collector output