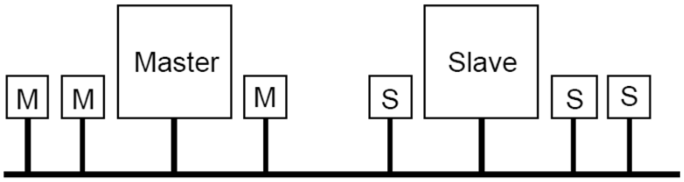
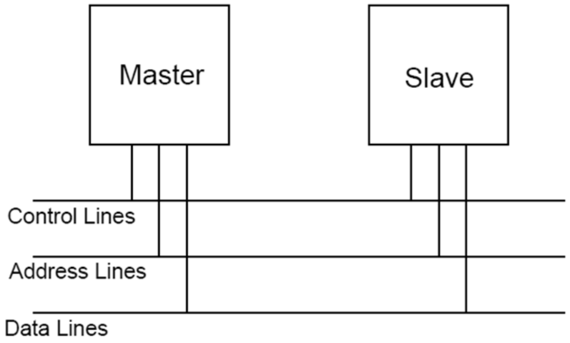
Bus Basics - Classic

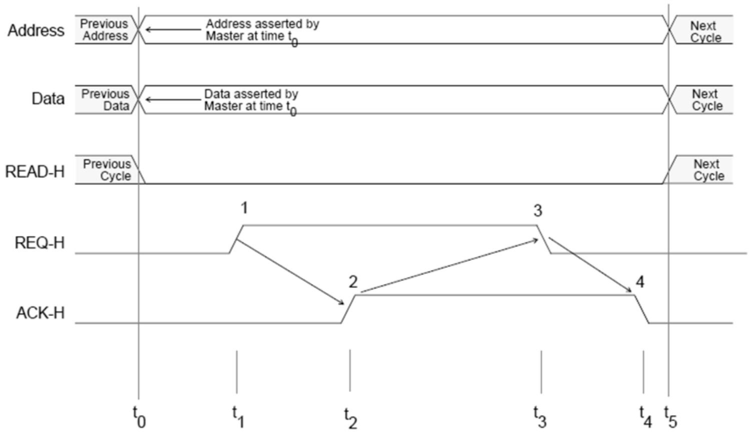
Conceptual Level Block Diagram



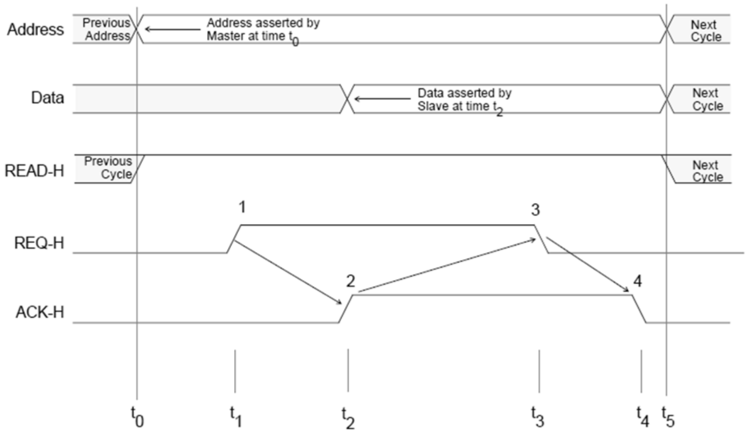
Two Module Model



Classic Write Sequence

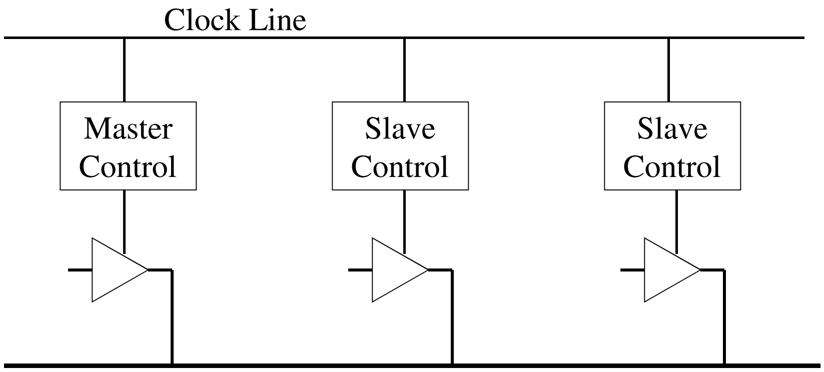


Classic Read Sequence



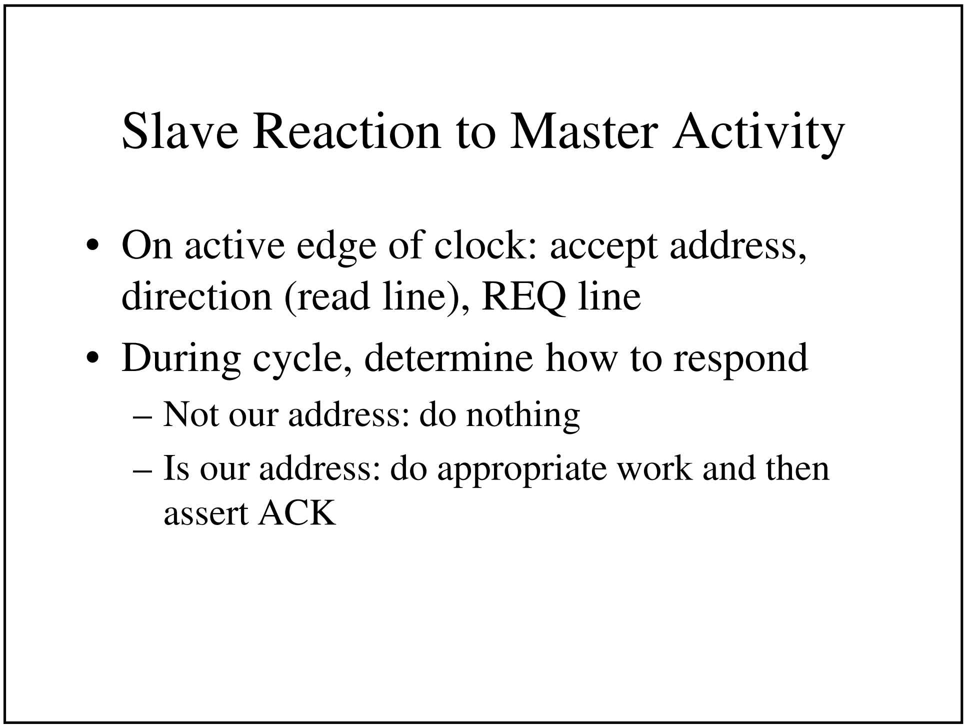
Bus Transactions - Clocked

Basic Concept: Master to Slave with Common Clock



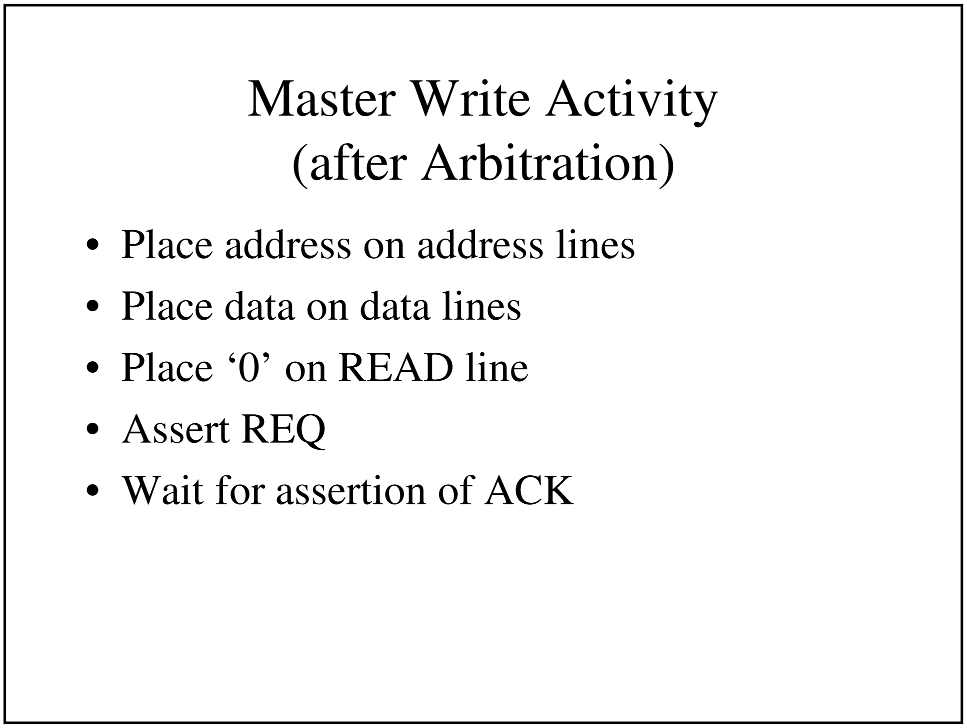
Master Begins Process:

- 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

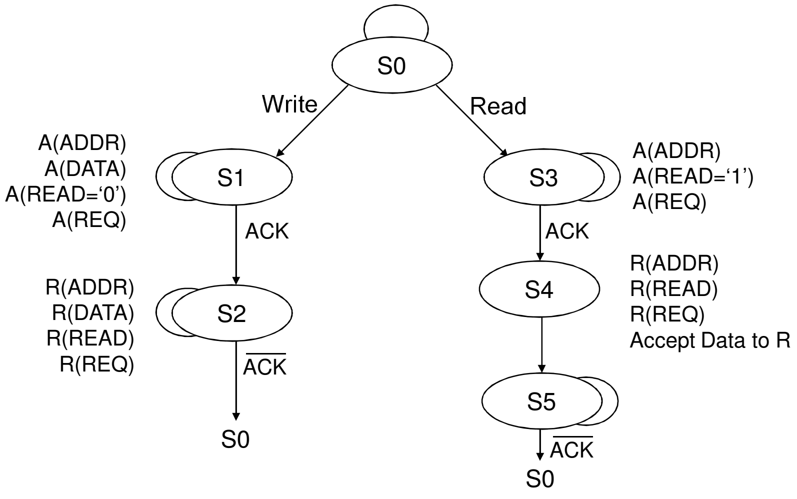
Master Read Activity (Termination)

- 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

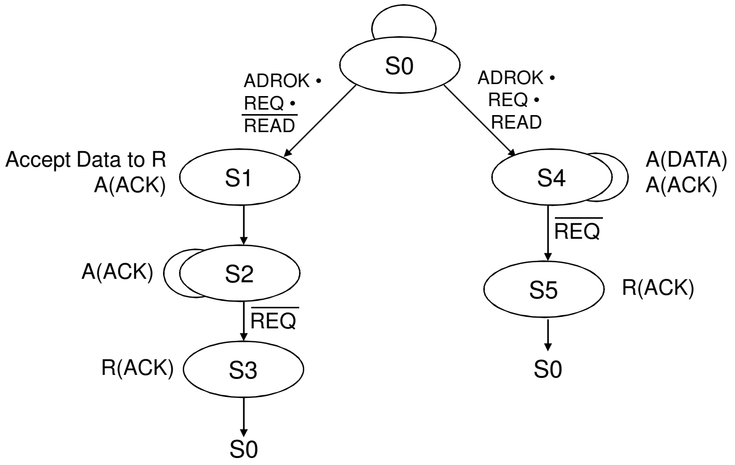
Slave Activity – Terminating Read Transfer

- 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


S0
ADROK •
REQ •
$\overline{READ}$
ADROK •
REQ •
READ
Accept Data to R
A(ACK)
S1
S4
A(DATA)
A(ACK)
$\overline{REQ}$
A(ACK)
S2
S5
R(ACK)
$\overline{REQ}$
R(ACK)
S3
S0
S0