Hardware Acceleration of Market Order Decoding

Amandeep Chhabra, Adil Sadik, Manu Dhundi,Prabhat Godse

Outline

- What is Hardware Acceleration of Market Order Decoding (HAMOD)
- Motivation
- System Overview
- Hardware
  - Read/Write, Multiplexer
  - Controller
- Software
- Results
- Acknowledgement

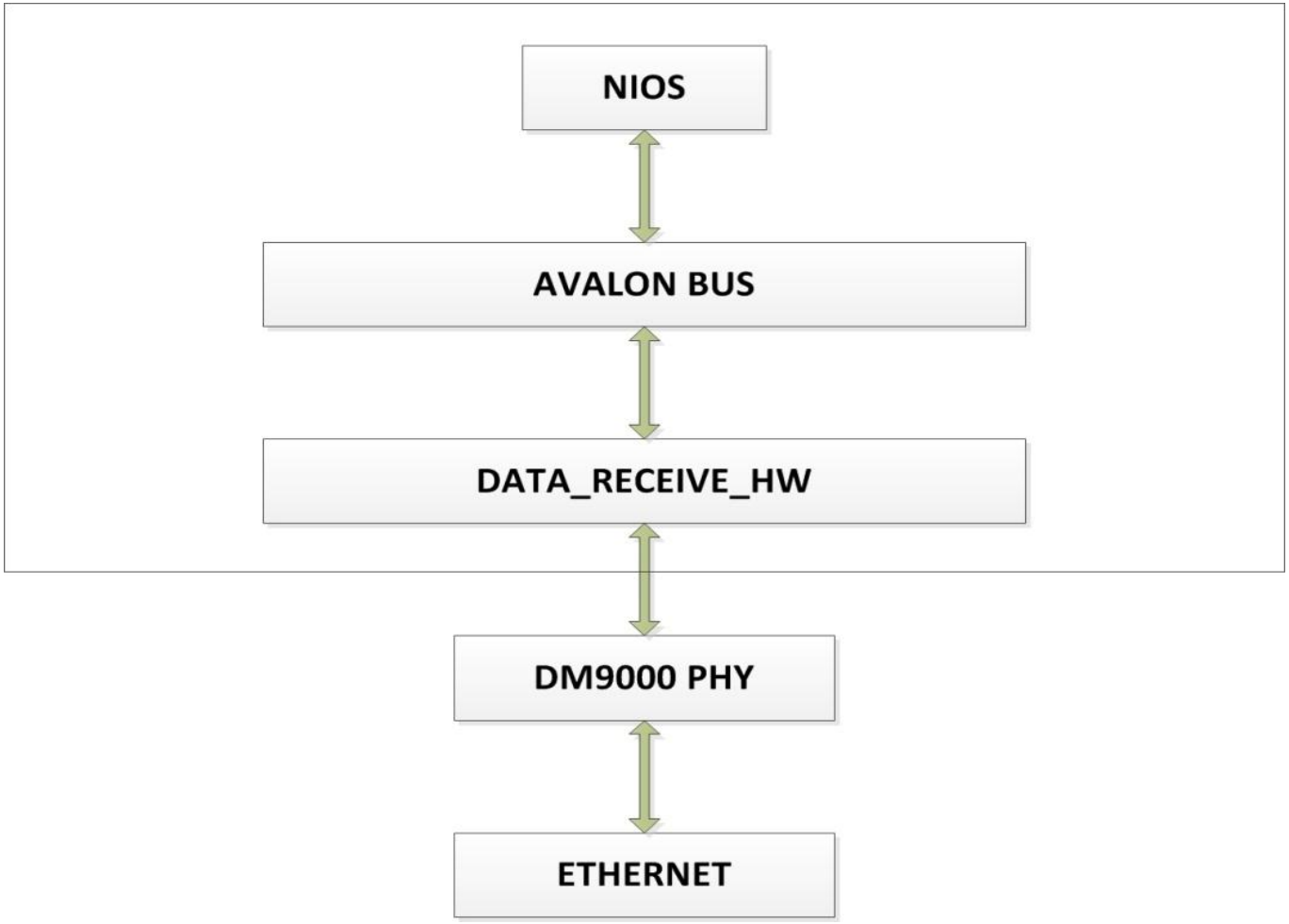
What is a HAMOD

- Accelerate the reading of Market Order Data using FPGA.
- Decrease the latency involved in reading data from Ethernet.
- Market data order similar to NASDAQ standard.
- UDP packets.
- Software processes orders and makes sample deals.

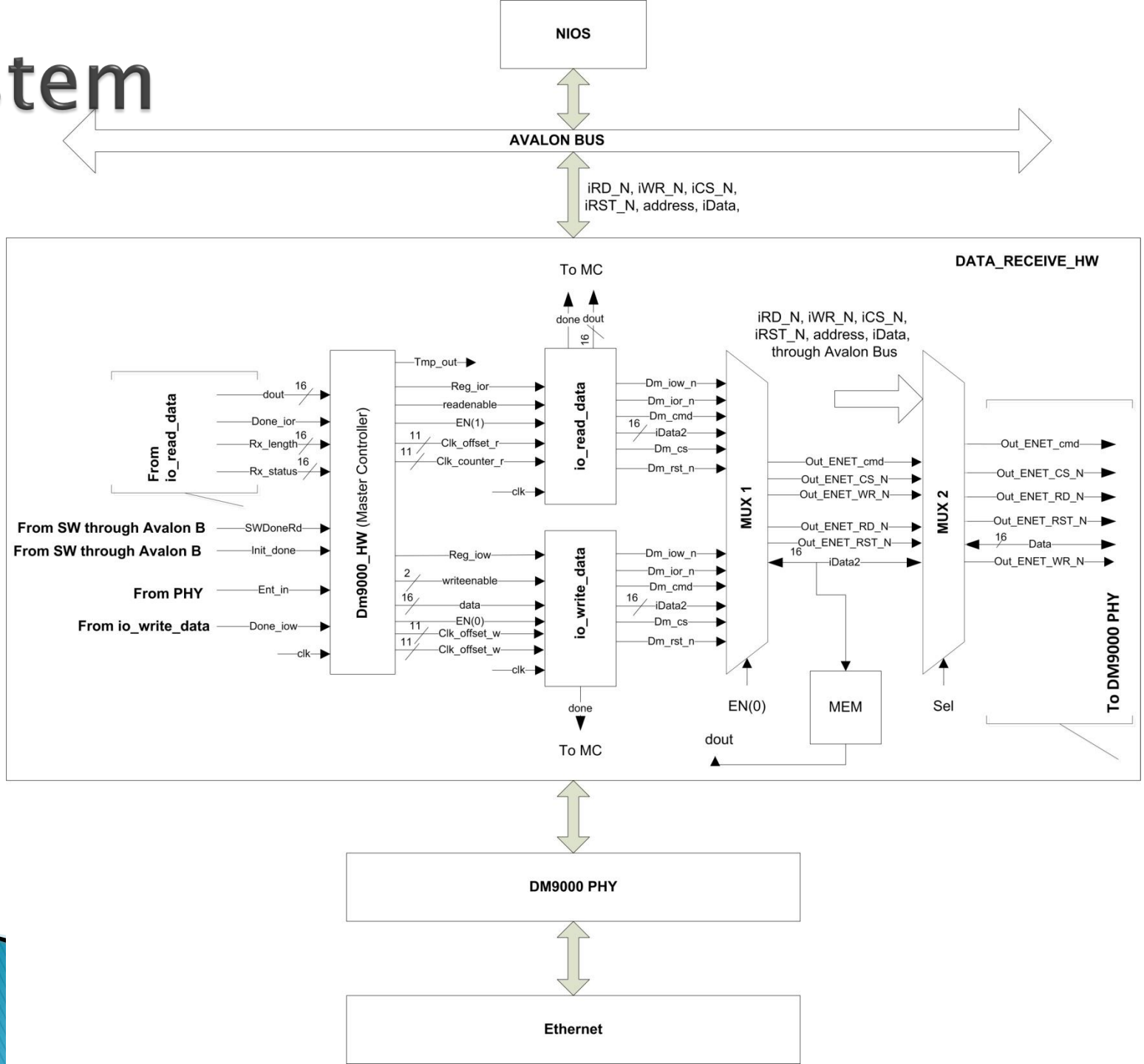
Motivation

- Low latency network systems
  - Application in finance
  - Data Centers
- Reconfigurable hardware systems
- Application in current industry.

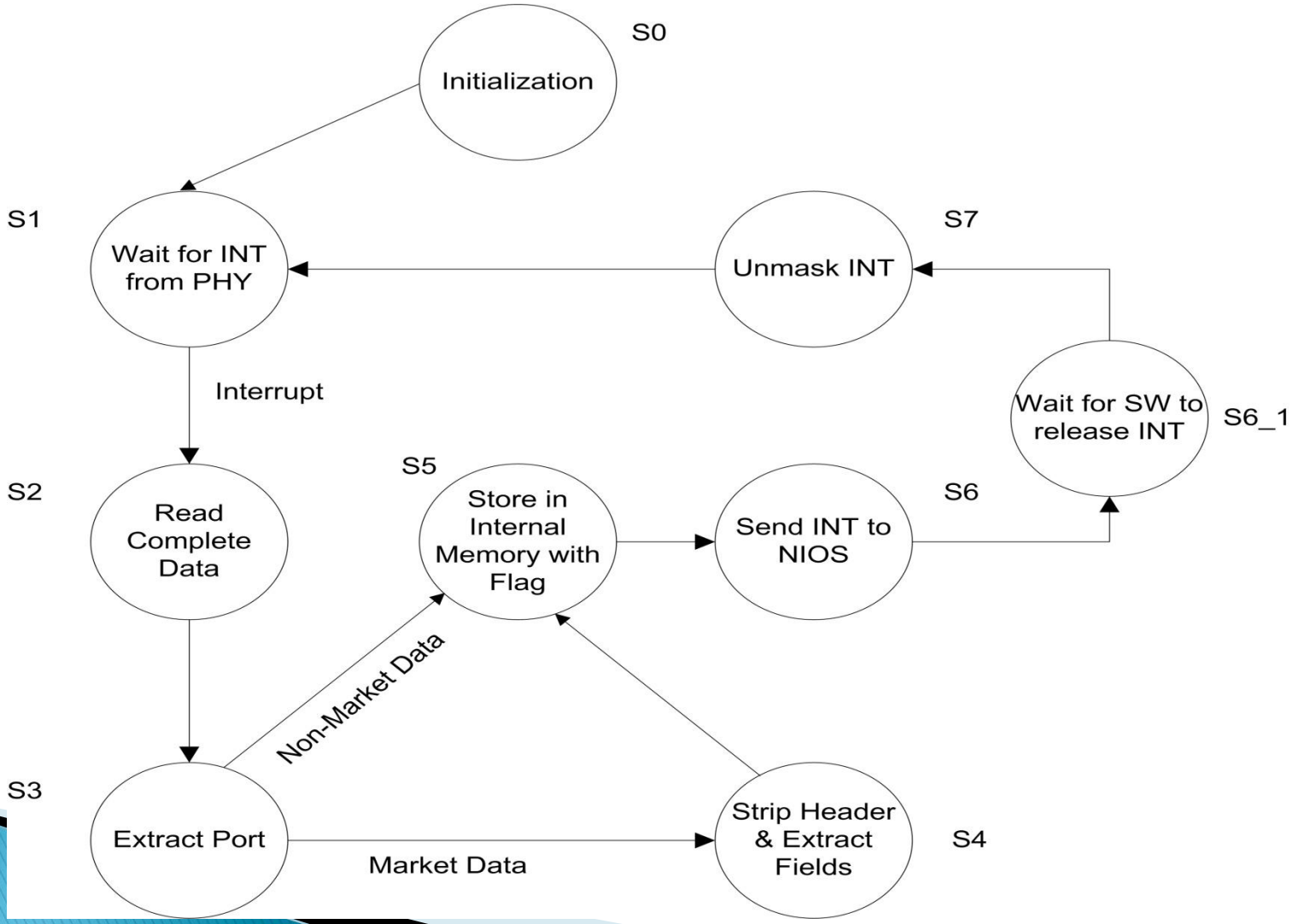
System



System

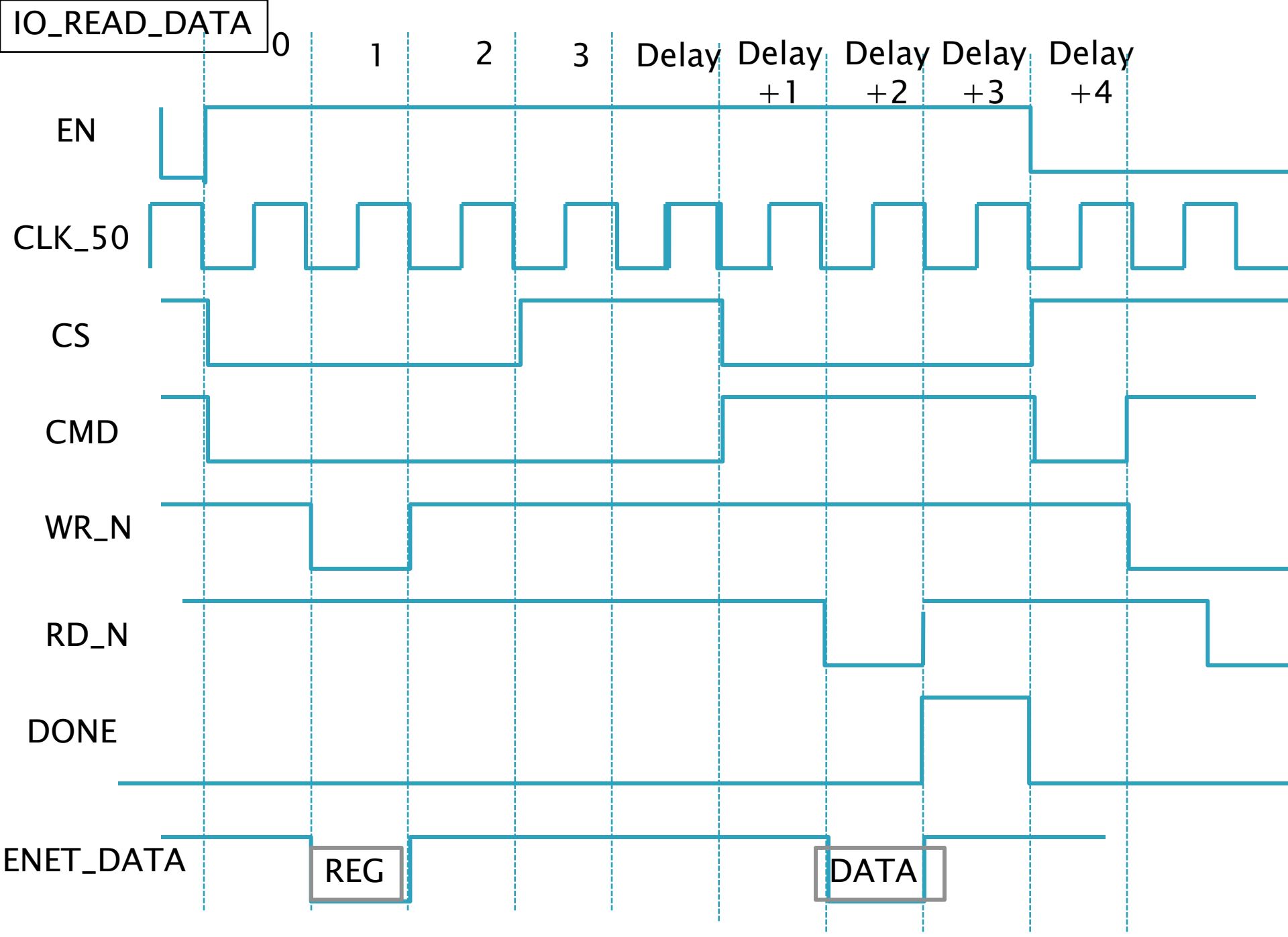

NIOS
AVALON BUS
iRD_N, iWR_N, iCS_N,
iRST_N, address, iData,
DATA_RECEIVE_HW
To MC
done dout
16
iRD_N, iWR_N, iCS_N,
iRST_N, address, iData,
through Avalon Bus
From
io_read_data
dout
Done_ior
Rx_length
Rx_status
16
Dm9000_HW (Master Controller)
Tmp_out
Reg_ior
readenable
EN(1)
11
Clk_offset_r
11
Clk_counter_r
clk
io_read_data
Dm_iow_n
Dm_ior_n
Dm_cmd
16
iData2
Dm_cs
Dm_rst_n
From SW through Avalon B
SWDoneRd
From SW through Avalon B
Init_done
From PHY
Ent_in
From io_write_data
Done_iow
clk
Reg_iow
2
writeenable
16
data
EN(0)
11
Clk_offset_w
11
Clk_offset_w
clk
io_write_data
Dm_iow_n
Dm_ior_n
Dm_cmd
16
iData2
Dm_cs
Dm_rst_n
done
To MC
MUX 1
Out_ENET_cmd
Out_ENET_CS_N
Out_ENET_WR_N
Out_ENET_RD_N
Out_ENET_RST_N
16
iData2
MUX 2
Out_ENET_cmd
Out_ENET_CS_N
Out_ENET_RD_N
Out_ENET_RST_N
16
Data
Out_ENET_WR_N
To DM9000 PHY
EN(0)
MEM
Sel
dout
DM9000 PHY
Ethernet

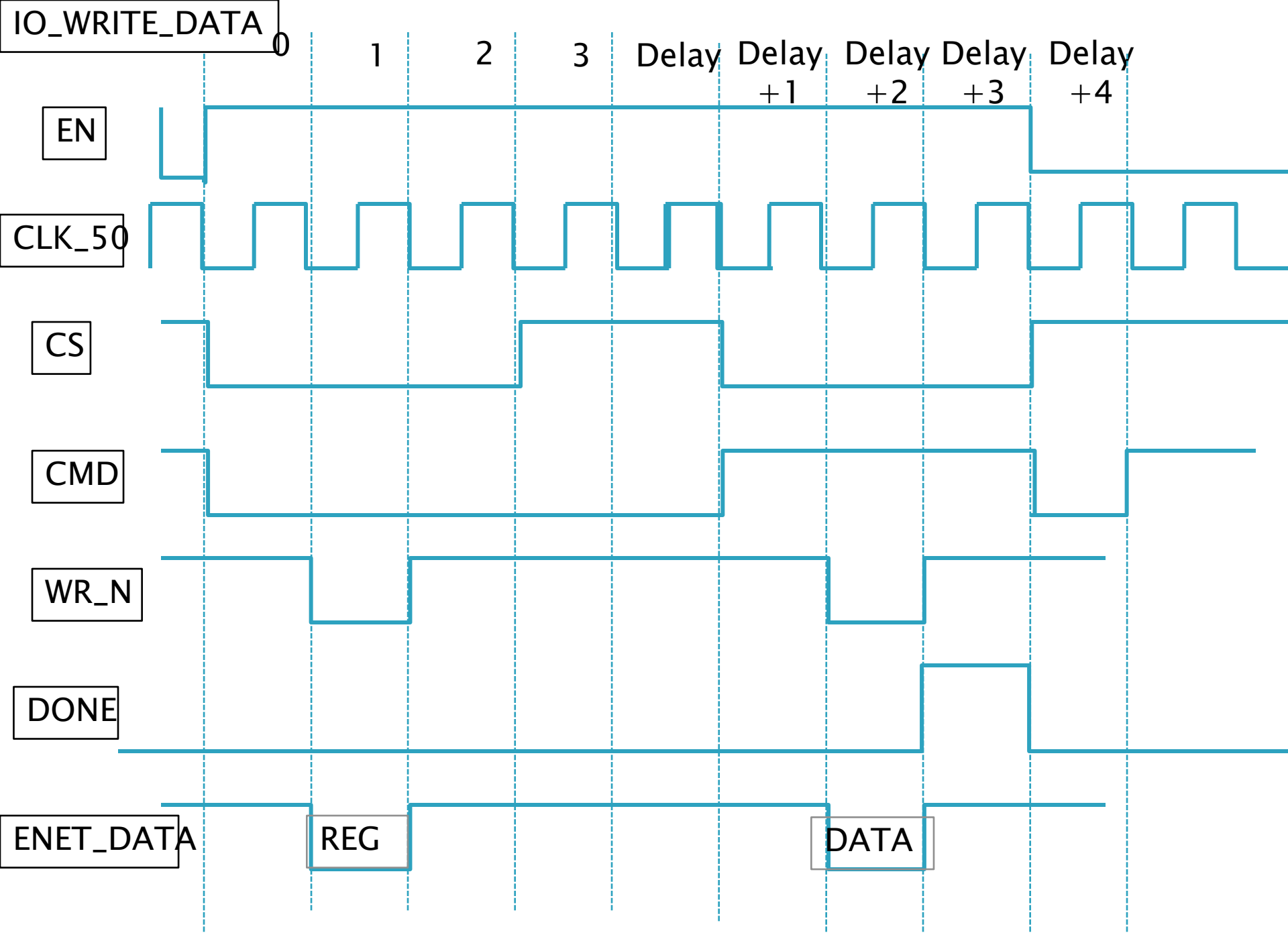
System


S0
Initialization
S1
Wait for INT from PHY
S7
Unmask INT
Interrupt
Wait for SW to release INT
S6_1
S2
Read Complete Data
S5
Store in Internal Memory with Flag
S6
Send INT to NIOS
Non-Market Data
S3
Extract Port
Market Data
Strip Header & Extract Fields
S4

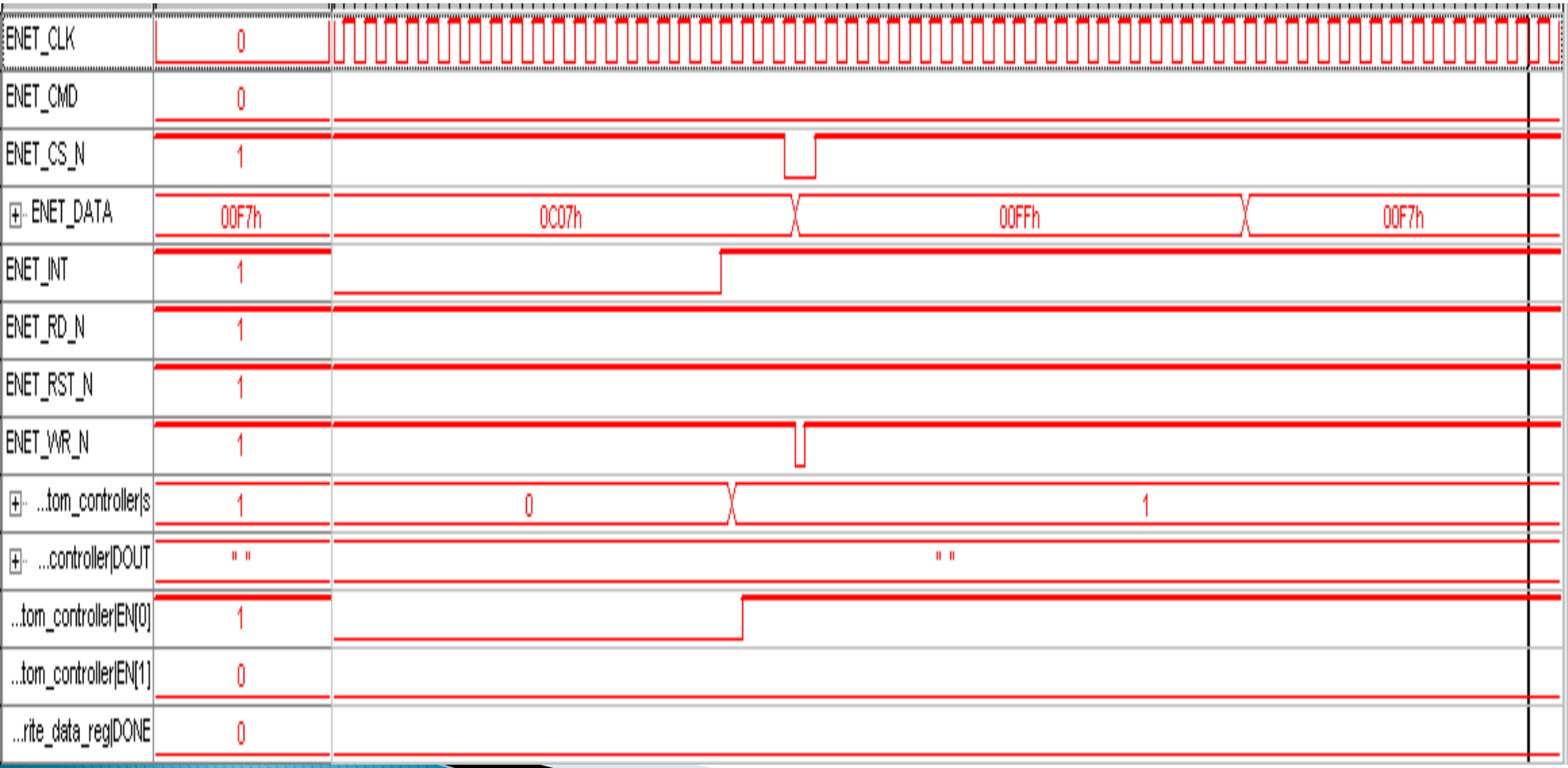
Hardware

- IO_Read with timing Diagram
- IO_Write with timing Digram


IO_READ_DATA
0
1
2
3
Delay
Delay +1
Delay +2
Delay +3
Delay +4
EN
CLK_50
CS
CMD
WR_N
RD_N
DONE
ENET_DATA
REG
DATA

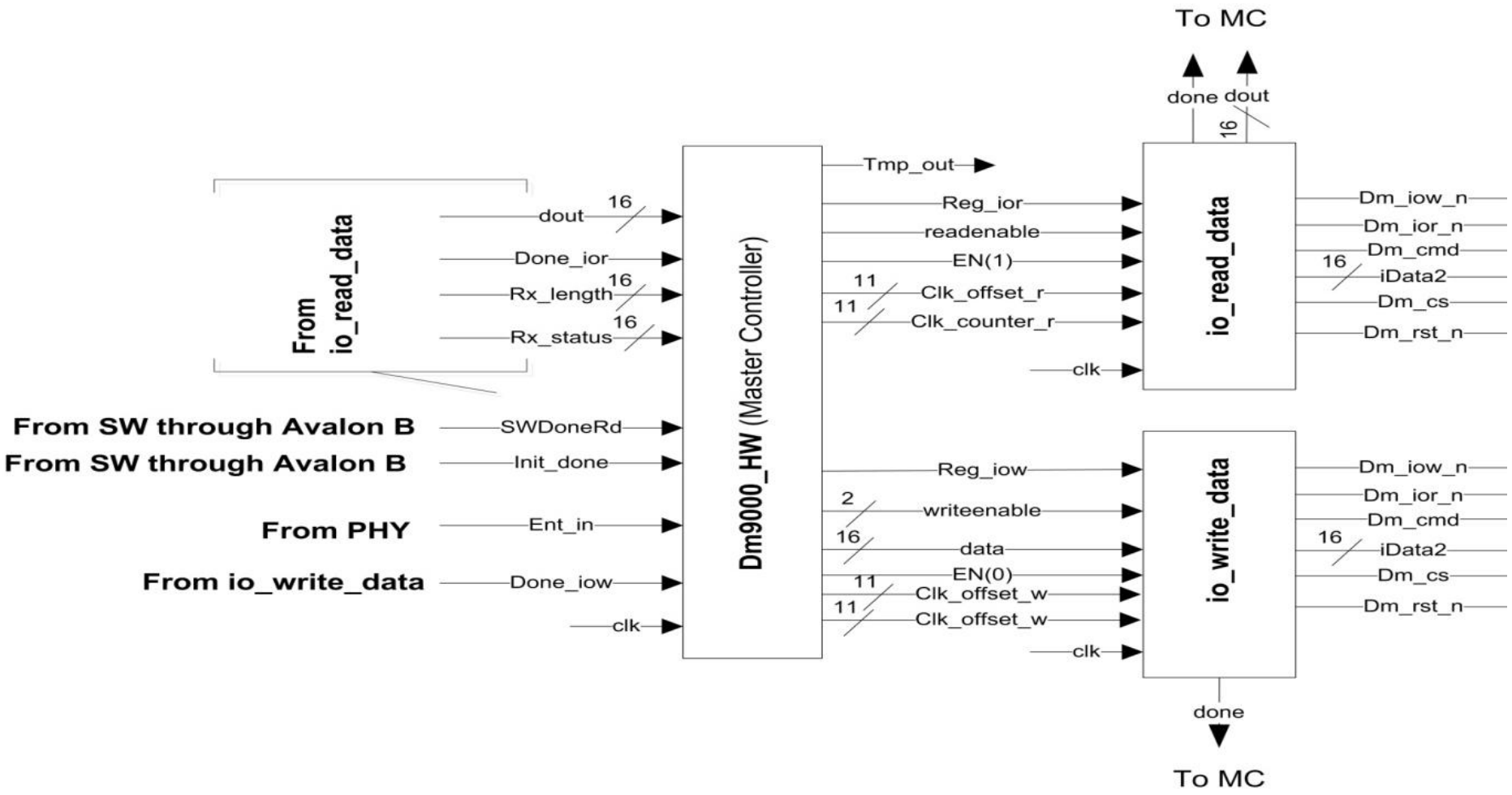

IO_WRITE_DATA
0
1
2
3
Delay
Delay +1
Delay +2
Delay +3
Delay +4
EN
CLK_50
CS
CMD
WR_N
DONE
ENET_DATA
REG
DATA

SignalTap Analyzer

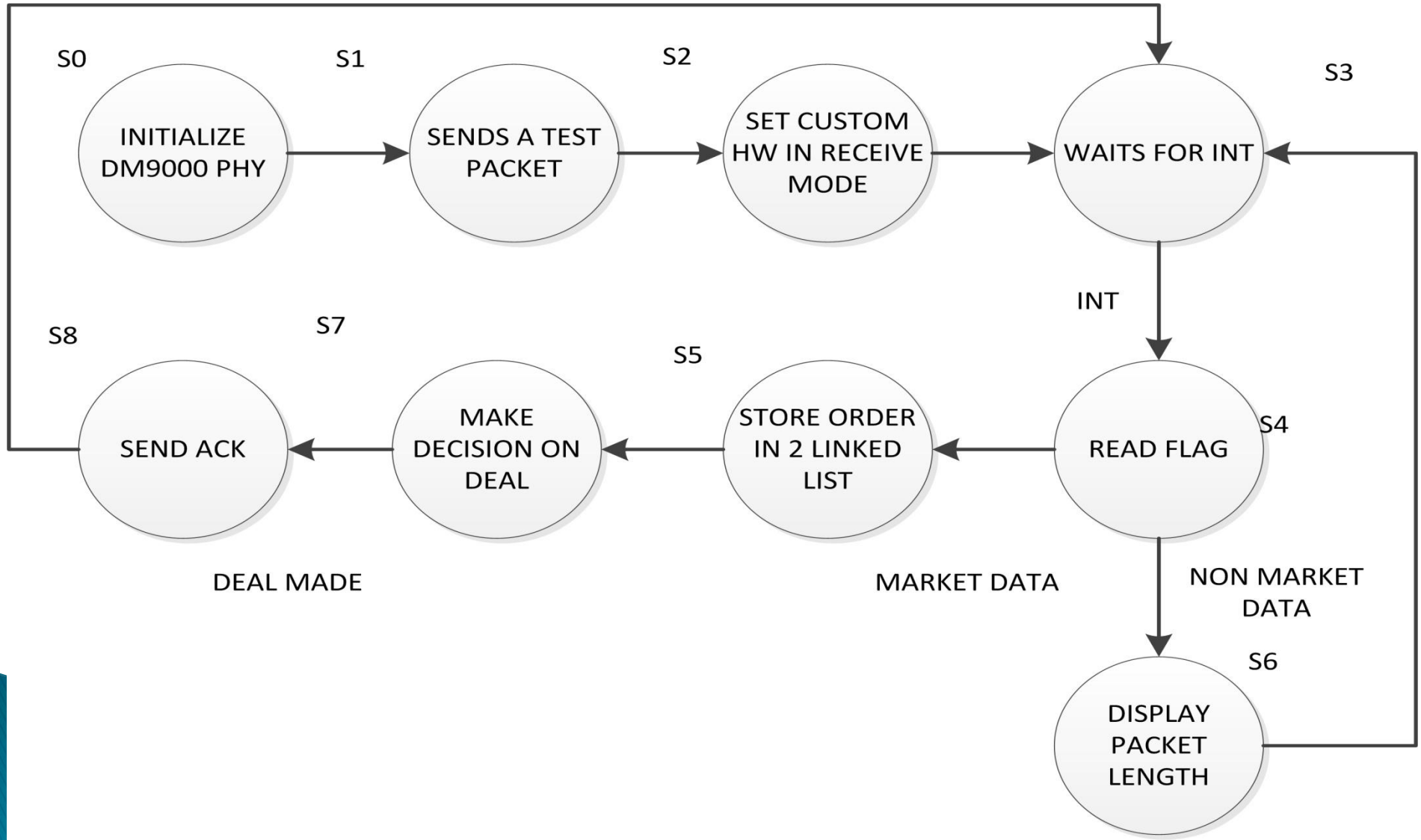


Hardware(conti)

- Controller



Software


S0
INITIALIZE DM9000 PHY
S1
SENDS A TEST PACKET
S2
SET CUSTOM HW IN RECEIVE MODE
S3
WAITS FOR INT
INT
S4
READ FLAG
S5
STORE ORDER IN 2 LINKED LIST
S7
MAKE DECISION ON DEAL
S8
SEND ACK
DEAL MADE
MARKET DATA
NON MARKET DATA
S6
DISPLAY PACKET LENGTH

Result/Performance Benchmark

- Successfully receive Market Order Packet and Extract the fields.
- Clock Cycles for HW read and decode=982
- Clock Cycles for HW + SW read and decode: 14562
- Clock Frequency 50MHz
- We are off-course better than lab2.

Hurdles

- On-chip debugging
- Interrupt management
- Software Memory Constraints
- Poor documentation of DM9000a PHY

Special Thanks

- Professor Stephen Edwards
- David Lariviere (TA)

Questions

???