Introduction to Barefoot

May 2011
Hyunmo Chung
CTO
Contents

• Jasmine Platform Overview
• Block Diagram of Barefoot
• NAND Flash Configuration
• NAND Flash Timing
• NAND Flash Controller
• Flash Operation Parallelism
• Buffer Management
• SATA Command Queuing
• DRAM ECC and Addressing Issue
Jasmine Platform Overview

• **Barefoot Controller**
  – ARM7TDMI-S CPU up to 87.5MHz
  – SATA 3Gbps
  – Mobile SDRAM controller up to 175MHz, up to 64MB
  – NAND flash BCH 8/12/16 bit correction per sector
  – SDRAM Reed Solomon 2 byte correction per 128 + 4 byte
  – Maximum 64 CE’s (4 channels, 16 bit/ch, 8 bank/ch)

• **Mobile SDRAM: Samsung 64MB, subject to change**

• **NAND Flash: Samsung 64GB, subject to change**
  – 35nm MLC, asynchronous mode
  – 32Gb (4GB) per die, Dual Die Package, 2 CE signals per package, 8 packages

• **Debugging/Monitoring aids**
  – JTAG
  – UART
  – 1 LED and 6 GPIO pins
  – Mictor connector to NAND flash signals for logic analyser
  – Separate current measurement points for core, IO, SDRAM and NAND
NAND Flash Configuration

- **Four channels**
  - Fully parallel and independent operation
  - 16 bit IO bus in each channel

- **Eight banks in each channel**
  - The banks share the same IO bus. → IO operations cannot be parallel.
  - However, cell operations can occur in parallel.
  - Since Barefoot has only four R/B signal inputs (each channel) from banks, max 4 way interleaving is possible.

- **Two 8-bit-flash-chips in each bank**
  - The lower byte of IO bus is for one chip and the higher is for the other.
  - Two chips receive the same CLE/ALE/WE/RE signals. → They receive the same command at the same time.
  - CE signals are separate, but they usually switch together most of the time. (cf. FO_H, FO_L)
  - A bank is considered idle if and only if the two chips are idle.
NAND Flash Timing

- Example of system clock = 175MHz
  period = 5.714ns
  set up length = 5.714 * N / 2, where N can be chosen from 2, 4, 5, 6, 7, 8, 9, ...
  hold length = 5.714 * M, where M can be chosen from 1, 2, 3, ...
- Given restrictions (Samsung 35nm async mode)
  set up minimum 15ns
  hold minimum 10ns
- N = 5, set up length = 5.714 * 2.5 = 14.29ns
- M = 2, hold length = 5.714 * 2 = 11.43ns
- Flash access cycle = set up + hold = 25.7ns
# NAND Flash Controller

## Flash Command Port
- CMD
- BANK
- OPTION
- DMA_ADDR
- DMA_CNT
- COLUMN
- ROW_0_L
- ROW_0_H
- ...
- ROW_31_L
- ROW_31_H
- DST_COL
- DST_ROW
- ISSUE

## Waiting Room
- CMD
- BANK
- OPTION
- ...

## Bank Status Ports
- CMD
- OPTION
- ...

### Process Flow:
1. Issue
2. Accept
3. Abort
Flash Operation Parallelism

BSP_FSM of Bank A0

| IDLE | BUSY | IDLE |

RB signal of Bank A0

| READY | BUSY (doing a cell operation) | READY |

BSP_FSM of Bank A1

| IDLE | BUSY | IDLE |

RB signal of Bank A1

| READY | BUSY (doing a cell operation) | READY |

Owner of Channel A (= who is doing an interface operation)

| FREE | A0 | A1 | FREE | A1 | FREE | A0 | FREE |
Buffer Management

- Buffer segmentation: 4~32KB per buffer (must be identical to virtual flash page size)
- Read buffer space and write buffer space are separate.
- Circular buffer scheme
- Buffer management and flow control is done by hardware
  - SATA write pointer does not run ahead of BM write limit
  - SATA read pointer does not run ahead of BM read limit
SATA Command Queuing

- SATA Event Queue
  - 128 slots for SATA commands
  - An entry is inserted by ISR upon command reception.
  - An entry is removed by FTL top level loop and processed.
  - Basically it is a FIFO queue. However, reads can have priority over writes.
- NCQ
  - 32 slots for SATA commands
  - FIFO policy
DRAM ECC and Addressing Issue

- Four byte ECC parity is added to every 128 bytes of data
- Firmware can only utilize 64MB * 128 / 132 = 65075200 bytes
- When CPU does a direct access (not via Memory Utility hardware) to DRAM, ECC engine is bypassed and the firmware should be aware of the existence of address holes.
- Recommendation for academic research
  - Enable DRAM ECC engine.
  - For writes, use Memory Utility so that parity codes are updated to reflect the new data.
  - For reads from DRAM to CPU, do direct access in order to avoid ECC overhead.
  - For reads from DRAM to Flash, ECC engine is used if it is enabled.
- Recommendation for commercial use
  - Never bypass ECC engine.
Thank You