ONFI 3.2 and Flash Control Basics
Open NAND Flash Interface—Basics

The command set

Addressing and activation

Figuring out what you’re attached to

Command overview, including concurrent accesses

NV-DDR and NV-DDR2 interfaces
Recall general device organization

- I/O control
- Address register
- Status register
- Command register
- Control logic
- Column decode
- NAND Flash array (2 planes)
- Data register
- Cache register
- Row decode

---

Command register
Address register
Status register
Control register

---

Row decode
Column decode
NAND Flash array
Data register
Cache register
Recall general device organization

Control signals (‘#’ = active low)
- CE# — Chip Enable
- CLE — Command Latch Enable
- ALE — Address Latch Enable
- WE# — Write Enable
- RE# — Read Enable
- WP# — Write Protect
- R/B# — Read/Busy (from SR[6])

There is also a CLK …
Three Interfaces:

SDR — what used to be called “asynchronous”
  — can be 8-bit or 16-bit
  — relatively slow

NV-DDR — used to be called “source-synchronous”
  — must be 8-bit
  — up to 100MHz, 200MT/s = 200MB/s

NV-DDR2 — must be 8-bit
  — up to 267MHz, 533MT/s = 533MB/s
Figure 31 shows an example of a Target memory organization. In this case, there are two logical units where each logical unit has two planes.

A device contains one or more targets. A target is controlled by one $CE_n$ signal. A target is organized into one or more logical units (LUNs). A logical unit (LUN) is the minimum unit that can independently execute commands and report status. Specifically, separate LUNs may operate on arbitrary command sequences in parallel. For example, it is permissible to start a Page Program operation on LUN 0 and then prior to the operation's completion to start a Read command on LUN 1. See multiple LUN operation restrictions in section 3.1.3.

A LUN contains at least one page register and a Flash array. The number of page registers is dependent on the number of multi-plane operations supported for that LUN. The Flash array contains a number of blocks.

Logical Unit 0

Logical Unit 1

“Page Register”
Several ways to group internal devices I

Shared buses, not ganged

Discrete CE per device example
Several ways to group internal devices II

CE Pin Reduction examples

CE Pin Reduction examples

Shared buses, not ganged
The page address always uses the least significant row address bits. The block address uses the middle row address bits and the LUN address uses the most significant row address bit(s).

3.1.1. Multi-plane Addressing

The multi-plane address comprises the lowest order bits of the block address as shown in Figure 33.

The following restrictions apply to the multi-plane address when executing a multi-plane command sequence on a particular LUN:

- The plane address bit(s) shall be distinct from any other multi-plane operation in the multi-plane command sequence.
- The page address shall be the same as any other multi-plane operations in the multi-plane command sequence.

The device may indicate multi-plane block address restrictions. The specific cases are:

- No restriction: All block address bits may be different between two plane addresses.
- Full restriction: All block address bits (other than the plane address bits) shall be the same between two plane addresses.
- Lower bit XNOR restriction: If the XNOR of the lowest plane address bits (bit 0) is one between two plane addresses, then there is a full restriction between these two plane addresses. If the XNOR of the lowest plane address bits is zero between two plane addresses, then there is no restriction between these two plane addresses.

Table 19 illustrates the three types of restrictions for a four plane operation.

<table>
<thead>
<tr>
<th>Restriction Type</th>
<th>Plane Address 0</th>
<th>Plane Address 1</th>
<th>Plane Address 2</th>
<th>Plane Address 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>No restriction</td>
<td>Block A</td>
<td>Block B</td>
<td>Block C</td>
<td>Block D</td>
</tr>
<tr>
<td>XNOR restriction</td>
<td>Block A+2</td>
<td>Block B+2</td>
<td>Block A+2</td>
<td>Block B+2</td>
</tr>
<tr>
<td>Full restriction</td>
<td>Block A+1</td>
<td>Block A+2</td>
<td>Block A+2</td>
<td>Block A+3</td>
</tr>
</tbody>
</table>

Table 20 describes whether there is a lower bit XNOR restriction between two plane addresses A and B, based on their plane address bits for a 4-plane implementation. If there is a lower bit XNOR restriction, then the block addresses (other than the plane address bits) shall be the same between multi-plane addresses A and B.

Note: Least significant column bit must be 0 for DDR interfaces.
This is easy

This is not

MC

CTL bus

IO bus

MC
Config I: Discrete CE per Device

Channel 0

CE0_n
Host Target 0
NAND Target 0
Volume H0N0
ENi_0
NC

CE1_n
Host Target 1
NAND Target 1
Volume H1N1
ENo_0
NC

CE2_n
Host Target 2
NAND Target 2
Volume H2N2
ENi_1
Channel 1

CE3_n
Host Target 3
NAND Target 3
Volume H3N3
ENo_1
NC

Target 0
Host
NC

Target 1
Host
NC

Target 2
Host
NC

Target 3
Host
NC

NAND Target 0
NAND Target 1
NAND Target 2
NAND Target 3
Config I: Discovery

Foreach distinct CE pin i, in order from CE_0 to CE_7:

Foreach distinct 8-bit bus j (could be up to 4):

Pull CE_i low

Using I/O Bus j:
Issue Reset (FFh) command
Issue Read ID (90h) command with address 20h

If ONFI signature is returned
   Make a note that CE_i + Bus j is valid combo
Next CE pin
else …
CE_n pin reduction is not supported in all topologies. If two NAND Targets in the same NAND package share the same data bus, then each shall expose a separate CE_n pin external to the NAND package. In this case, the host shall use distinct Host Targets (CE_n signals) with each of these NAND Targets.

The state of ENi determines whether the NAND package is able to accept commands. ENi is pulled high internal to the NAND package. If the ENi pin is high and CE_n is low for the NAND Target, then the NAND Target shall accept commands. If the ENi pin is low for the NAND Target, then the NAND Target shall not accept commands.

Note: The first command issued after a power-on is a special case, refer to the initialization sequence in section 3.5.2.

ENO is driven low by the device when CE_n is low and a Volume address is not appointed for the NAND Target. ENo is tri-stated by the device when the CE_n associated with the NAND Target is low and a Volume address is appointed for that NAND Target. When the CE_n signals for all NAND Targets that share an ENo signal are high, ENo is tri-stated by the device. Note that ENo is...
Config II: Discovery

1. Power is applied to the NAND device(s).

2. CE_n (Host Target) is pulled low.

3. If resetting all NAND Targets in parallel, then the host issues the Reset (FFh) command. This command is accepted by all NAND Targets connected to the CE_n (Host Target).

4. If resetting each NAND Target sequentially, then:
   a. Host issues Read Status (70h) command. Issuing Read Status (70h) prior to any other command indicates sequential Reset (FFh) of each NAND Target.
   b. Host issues Reset (FFh). This command only resets the NAND Target connected to the CE_n (Host Target) whose ENi signal is high.

5. Host issues Read Status (70h) command and waits until SR[6] is set to one.

6. Host configures the NAND Target. Read ID, Read Parameter Page, and other commands are issued as needed to configure the NAND Target.

7. Set Feature with a Feature Address of Volume Configuration is issued to appoint the Volume address for the NAND Target(s) whose ENi signal is high. The Volume address specified shall be unique amongst all NAND Targets. After the Set Features command completes, ENo is pulled high and the Volume is deselected until a Volume Select command is issued that selects the Volume. The host shall not issue another command to a NAND Target connected to the associated Host Target until after tFEAT time has elapsed.

8. For each NAND Target connected to the Host Target, steps 4-7 are repeated for the sequential initialization sequence and steps 5-7 for the parallel initialization sequence.

9. When no further NAND Targets are found connected to the Host Target, then repeat steps 2-8 for the next Host Target (i.e. host CE_n signal).

10. To complete the initialization process, a Volume Select command is issued following a CE_n transition from high to low to select the next Volume that is going to execute a command.
Config II: Discovery

Basic idea:

Go through and assign a different Volume Address (via Set Features) to each device on the same CE line

Thereafter, each time you want to execute a given command (Read, Program, Erase, etc.) you need to pull CE low and execute a Volume Select command first
## ONFI Cmd Set

### Read
- **Read**
- **Change Read Column**
- **Read Cache**
- **Page Program**
- **Page Cache Program**
- **Block Erase**
- **Read Status**
- **Read ID**
- **Read Parameter Page**

### Table 69: Command set

<table>
<thead>
<tr>
<th>Command</th>
<th>O/M</th>
<th>1st Cycle</th>
<th>2nd Cycle</th>
<th>Acceptable while Accessed LUN is Busy</th>
<th>Acceptable while Other LUNs are Busy</th>
<th>Target level commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>M</td>
<td>00h</td>
<td>30h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-plane</td>
<td>O</td>
<td>00h</td>
<td>32h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copyback Read</td>
<td>O</td>
<td>00h</td>
<td>35h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Read Column</td>
<td>M</td>
<td>05h</td>
<td>E0h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Read Column Enhanced</td>
<td>O</td>
<td>06h</td>
<td>E0h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Cache Random</td>
<td>O</td>
<td>00h</td>
<td>31h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Cache Sequential</td>
<td>O</td>
<td>31h</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Cache End</td>
<td>O</td>
<td>3Fh</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Block Erase</td>
<td>M</td>
<td>60h</td>
<td>D0h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-plane</td>
<td>O</td>
<td>60h</td>
<td>D1h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Status</td>
<td>M</td>
<td>70h</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Status Enhanced</td>
<td>O</td>
<td>78h</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page Program</td>
<td>M</td>
<td>80h</td>
<td>10h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-plane</td>
<td>O</td>
<td>80h</td>
<td>11h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page Cache Program</td>
<td>O</td>
<td>80h</td>
<td>15h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copyback Program</td>
<td>O</td>
<td>85h</td>
<td>10h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-plane</td>
<td>O</td>
<td>85h</td>
<td>11h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Small Data Move`</td>
<td>O</td>
<td>85h</td>
<td>11h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Write Column`</td>
<td>M</td>
<td>85h</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Row Address`</td>
<td>O</td>
<td>85h</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read ID</td>
<td>M</td>
<td>90h</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Volume Select`</td>
<td>O</td>
<td>E1h</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ODT Configure`</td>
<td>O</td>
<td>E2h</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Parameter Page</td>
<td>M</td>
<td>ECh</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Unique ID`</td>
<td>O</td>
<td>E0h</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Get Features`</td>
<td>O</td>
<td>EEh</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set Features`</td>
<td>O</td>
<td>EFh</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUN Get Features`</td>
<td>O</td>
<td>D4h</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUN Set Features</td>
<td>O</td>
<td>D5h</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reset LUN</td>
<td>O</td>
<td>FAh</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronous Reset</td>
<td>O</td>
<td>FCh</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reset</td>
<td>M</td>
<td>FFh</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:**
1. Change Write Column specifies the column address only. Change Row Address specifies the row address and the column address. Refer to the specific command definitions.
2. Small Data Move’s first opcode may be 80h if the operation is a program only with no data output. For the last second cycle of a Small Data Move, it is a 10h command to confirm the Program or Copyback operation.
3. Volume Select shall be supported if the device supports either CE_n pin reduction or matrix termination. ODT Configure shall be supported if the device supports matrix termination.
Sending a Command (NV-DDR)

- CLK
- CE_n
- CLE
- ALE
- W/R_n
- DQ[7:0]
- DQS

Figure 75 NV-DDR data interface command description
Data Input to Flash Device

- CE_n: Chip Enable
- CLE: Chip Enable Low
- ALE: Address Enable
- CLK: Clock
- W/R_n: Write/Read
- DQS: Data strobe enable
- DQ[7:0]: Data output

Timing parameters:
- tCKL: Clock low-to-high transition
- tCKH: Clock high-to-low transition
- tCS: Chip select active
- tCAD: Command active duration
- tCAL: Command active duration
- tDS: Data setup time
- tDQSS: Data strobe setup time
- tWPRE: Write precharge time
- tWPST: Write post-stabilization time
- tDQSH: Data strobe hold time
- tDQSL: Data strobe loss time
- tDSS: Data stabilization time
- tDSH: Data stabilization hold time
- tDQSL: Data strobe loss time
- tCAD starts for next non-idle cycle
Data Output from Flash Device

- CE_n
- CLE
- ALE
- CLK
- W/R_n
- DQS
- DQ[7:0]

Don't Care
Data Transitioning
Device Driving

ICAD starts for next non-idle cycle
Read ID

Address 00h — Manufacturer & Device IDs (2 bytes)

Cycle Type

<table>
<thead>
<tr>
<th>Cycle Type</th>
<th>CMD</th>
<th>ADDR</th>
<th>DOUT</th>
<th>DOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>90h</td>
<td>00h</td>
<td>MID</td>
<td>DID</td>
</tr>
</tbody>
</table>

R/B_n

Address 20h — ‘O’ ‘N’ ‘F’ ‘I’ (4 bytes)

Cycle Type

<table>
<thead>
<tr>
<th>Cycle Type</th>
<th>CMD</th>
<th>ADDR</th>
<th>DOUT</th>
<th>DOUT</th>
<th>DOUT</th>
<th>DOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>90h</td>
<td>20h</td>
<td>4Fh</td>
<td>4Eh</td>
<td>46h</td>
<td>49h</td>
</tr>
</tbody>
</table>

R/B_n
Read Parameter Page

Like Read ID, but returns 256-byte Parameter Page +more

Revision number, features supported
Manufacturer, product codes
Data bytes per page, spare bytes per page
Pages per block, blocks per LUN, number of LUNs
Number of address cycles: row and column
Number of bits per cell, number of planes
Endurance info, bad block info, ECC info
SDR, DDR, DDR2 timing mode support
Electrical parameters, timing parameters, driver strength
CRC bits at end
... plus, redundant copies of everything
Read Status

Returns the status of the last command processed

<table>
<thead>
<tr>
<th>Value</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status Register</td>
<td>WP_n</td>
<td>RDY</td>
<td>ARDY</td>
<td>VSP</td>
<td>CSP</td>
<td>R</td>
<td>FAILC</td>
<td>FAIL</td>
</tr>
</tbody>
</table>

FAIL        last command failed
FAILC       previous command failed
R           reserved
CSP         command-specific
VSP         vendor-specific
ARDY        Array Ready (if 1, no array operation in progress)
RDY         if 1, LUN is ready for a command
WP_n        write-protect (active low)

“Enhanced” version specifies row address
**Block Erase**

R1…R3 is row address (identifies device, block, page)

Successful if SR[0] (FAIL) is zero afterward

Host may not attempt to erase a bad block
(i.e., it is the responsibility of the controller to know bad-block information & keep it up to date)
Read Page

C1…C3 is column address; R1…R3 is row address

After tR (array-to-register), data is valid to be read out; controller should check the validity before bus transfer (via Read Status Enhanced)

Second issuance of 00h starts data read-out on IO bus
**Change Read Column (Enhanced)**

**USE:** Read Page command issued to LUN 0

Read Page command issued to LUN 1

Read Status Enhanced selects LUN 0

Change Read Column (Enhanced) issued to LUN 0

Data transferred from LUN 0

Read Status Enhanced selects LUN 1

Change Read Column (Enhanced) issued to LUN 1

Data transferred from LUN 1
Read Cache (Random, Sequential, End)

Read Cache Random (31h):

As defined for Read

Similar to previous example, but to the same LUN

As defined for

Read

CMD

ADDR

ADDR

ADDR

ADDR

ADDR

CMD

DOUT

DOUT

DOUT

DQx

30h

00h

C1

C2

R1

R2

R3

31h

DOUT

DOUT

DOUT

SR[6]

CMD

ADDR

ADDR

ADDR

ADDR

ADDR

CMD

DOUT

DQx

00h

C1

C2

R1

R2

R3

31h

D0

SR[6]
Read Cache Sequential (31h without address):

As defined for Read

Cycle Type

```
<table>
<thead>
<tr>
<th>Cycle Type</th>
<th>CMD</th>
<th>CMD</th>
<th>DOUT</th>
<th>DOUT</th>
<th>DOUT</th>
<th>CMD</th>
<th>DOUT</th>
</tr>
</thead>
</table>
```

DQx

```
<table>
<thead>
<tr>
<th>DQx</th>
<th>30h</th>
<th>31h</th>
</tr>
</thead>
</table>
```

SR[6]

```
<table>
<thead>
<tr>
<th>SR[6]</th>
<th>tWB</th>
<th>tR</th>
<th>tWB</th>
<th>tRFCBSY</th>
<th>tRR</th>
<th>tWB</th>
<th>tRFCBSY</th>
</tr>
</thead>
</table>
```

Read Cache End (3Fh):

As defined for Read Cache (Sequential or Random)

Cycle Type

```
<table>
<thead>
<tr>
<th>Cycle Type</th>
<th>CMD</th>
<th>DOUT</th>
<th>DOUT</th>
<th>DOUT</th>
<th>CMD</th>
<th>DOUT</th>
<th>DOUT</th>
</tr>
</thead>
</table>
```

DQx

```
<table>
<thead>
<tr>
<th>DQx</th>
<th>31h</th>
</tr>
</thead>
</table>
```

SR[6]

```
<table>
<thead>
<tr>
<th>SR[6]</th>
<th>tRFCBSY</th>
</tr>
</thead>
</table>
```

D0...Dn: Data bytes/words read from page requested by the previous cache operation.
Page Program

For operating one plane at a time

For operating two or more planes …
(to overlap multiple program operations in time)
Page Cache Program

Cycle Type: CMD ADDR ADDR ADDR ADDR ADDR ADDR DIN DIN DIN DIN DIN CMD

DQx: 80h C1 C2 R1 R2 R3 D0 D1 ... Dn 15h
SR[6] iPCBSY iWBS

Cycle Type: CMD ADDR ADDR ADDR ADDR ADDR ADDR DIN DIN DIN DIN DIN CMD

DQx: 80h C1 C2 R1 R2 R3 D0 D1 ... Dn 15h
SR[6] iPCBSY iWBS

Cycle Type: CMD ADDR ADDR ADDR ADDR ADDR ADDR DIN DIN DIN DIN DIN CMD

DQx: 80h C1 C2 R1 R2 R3 D0 D1 ... Dn 10h
SR[6] iPROG iWBS
Other interesting commands

**Copyback** (page copy within a single device)

**Small Data Move** (write less than a full page)

**Volume Select** (for addressing many devices on a CE)

**Get/Set Features** (to read/change a device’s parameters)

**Multi-plane Ops** (**Program, Copyback, Erase, Read**)  
*allows parallel or staggered commands to multiple planes for example …*
Multi-plane read timing

The Read command reads a page of data identified by a row address for the LUN specified. The page of data is made available to be read from the page register starting at the column address specified. With a multi-plane operation, multiple reads can be issued back to back to the LUN, with a shorter busy time between issuance of the next read operation.

Figure 125 defines the behavior and timings for issuing two multi-plane read commands. Figure 126 defines the behavior and timings for reading data after the multi-plane read commands are ready to return data.

Cache operations may be used when doing multi-plane read operations, as shown, if supported by the target as indicated in the parameter page. Refer to section 5.7.1.28.

Change Read Column Enhanced shall be issued prior to reading data from a LUN. If data is read without issuing a Change Read Column Enhanced, the output received is undefined.

---

**Slide 31**

---
NV-DDR2 Differences

Differential signaling (e.g., DQS_t and DQS_c) & ODT
CLK not used for command capture:

Optional Data Warm-up Cycles: