Improving Power Budgeting Estimates in NAND Applications

Michael Abraham
Applications Engineering Manager
Micron Technology, Inc.
(mabraham@micron.com)

Abstract

- Current NAND Flash Icc specifications do not give an accurate representation of active system power
- This presentation shows a better approach to measuring Icc that provides better predictability in the system of how much current devices will draw
- This methodology is particularly useful for battery powered applications like mobile phones, MP3 players, and GPS units as well as large-scale NAND solutions like SSDs

Power Budgeting

Accurate system power consumption is important for determining a power budget

- A power budget
  - Estimates system battery life
  - Determines voltage regulator sizes
  - Determines needed bulk capacitance
- An application-specific usage model is required to correctly calculate the power budget
  - Number of active NAND die
  - Single- vs. multi-plane operations
  - Active vs. Standby
  - Reads vs. writes
  - I/O vs. array

Original Icc Test Methodology

In current NAND Flash data sheets, there are few Icc definitions

- Read
- Program
- Erase
- Standby

Parameter	Conditions	Symbol	Typical	Max	Unit
Sequential read current	$^tRC = {}^tRC$ (MIN); CE# = Vil; Iout = 0mA	Icc1	20	30	mA
Program current	–	Icc2	20	30	mA
Erase current	–	Icc3	20	30	mA
Standby current	CE# = Vcc – 0.2V; WP# = 0V/Vcc	Isb	10	50	µA

Shortcomings of the Original Icc Test Methodology

- Icc parameters not sufficiently defined for an application-specific usage model
  - Current related to I/O and the data path should be separate from current related to array operations (program, read, erase)
  - Idle current is not defined
- Icc test methodology is not usable in high volume manufacturing (HVM)
  - Designed for single-site device characterization
  - Unable to correlate bench testing to HVM testing
- Icc test methodology is not reproducible
  - Icc1 (Read) implies data output as it shows Iout = 0mA
  - A program page operation requires data input, yet Icc2 shows no condition for it
  - Does Icc2 measurement include data input or only the current during ^{t}PROG?

Because of these shortcomings, NAND Flash Icc specifications do not give an accurate representation of active system power

Goals of an Icc Test Methodology

- HVM capable
- Reproducible
- Usable for application-specific power budgets
- Adaptable to multiple NAND interfaces
  - Asynchronous (up to 50 MB/s per x8 bus)
  - Synchronous DDR (up to 200 MB/s per x8 bus)

Icc Test Methodology Implementation Checklist

- Basic power-related device operations
- General test conditions
- Interface-specific test conditions
- Test sequences
- Formulas for correlation

Basic Power-related Operations

- Array read
- Array program
- Array erase
- I/O burst read (data output)
- I/O burst write (data input)
- Bus Idle
- Standby

All NAND behaviors should build on and be modeled from these basic operations

NAND Operation Examples

- Page Program is comprised of
  - Data input
  - Array program
  - Bus idle and/or standby
- Page Program Cache is comprised of
  - Data input
  - Array program + data Input
  - Bus idle and/or standby
- Between operations there is dead time
  - Bus idle – CE# remains LOW
  - Standby – CE# is pulled HIGH

Icc Parameters

Icc parameters added to allow application-specific power budget modeling

Parameter	Conditions	Symbol	Typ	Max	Unit
Array read current	See general and interface-specific test conditions	Icc1	20	30	mA
Array program current		Icc2	20	30	mA
Array erase current		Icc3	20	30	mA
I/O burst read current		Icc4r	20	30	mA
I/O burst write current		Icc4w	20	30	mA
Bus idle current		Icc5	2	3	mA
Standby current (CMOS)	CE# = Vccq – 0.2V; WP# = 0V/Vccq	Isb	10	50	µA

General Test Conditions

Parameter	Testing Condition
General conditions	1. Vcc = Vcc(min) to Vcc(max) 2. VccQ = VccQ(min) to VccQ(max) 3. CE# = 0V 4. WP# = VccQ 5. Iout = 0 mA 6. Measured across operating temperature range 7. N data input or data output cycles, where N is the number of bytes or words in the page 8. No interleaved operations 9. Sample 250 times at 1 millisecond intervals and average the results 10. Choose the first good even/odd block pair beginning at blocks 2-3
Array preconditioning for Icc1 and Icc3	The array is preconditioned to match the data input pattern for Icc2
Fixed wait time (no R/B# polling)	Icc1: ^{t}R = ^{t}R(max) Icc2: ^{t}PROG = ^{t}PROG(max) Icc3: ^{t}BERS = ^{t}BERS(max)

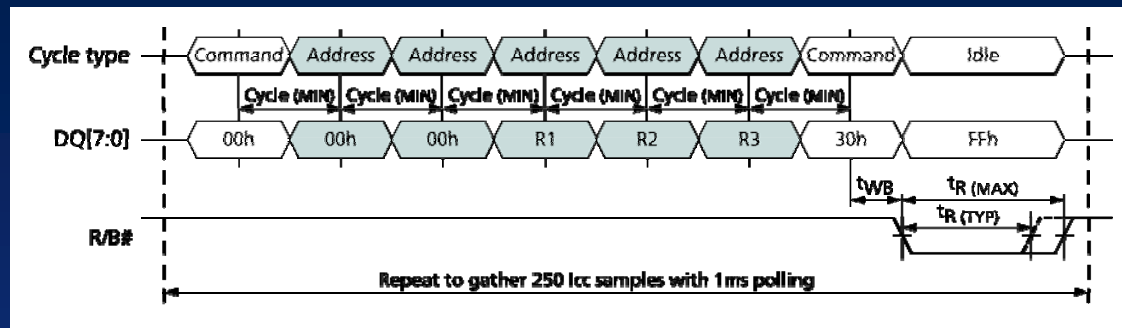
- Common across all NAND interfaces
- Fixed wait time is important for multi-site HVM testing

Define Interface-specific Test Conditions

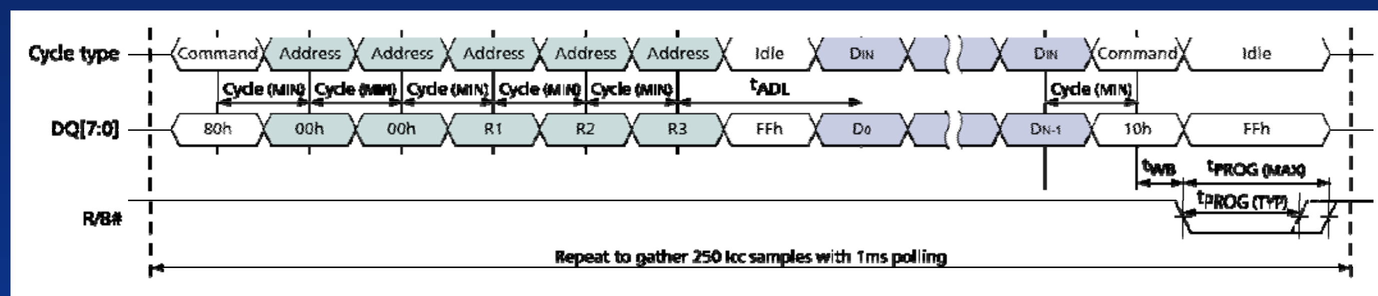
Parameter	Asynchronous	Synchronous DDR
AC Timing Parameters	$^{t}WC = {}^{t}WC(min)$ $^{t}RC = {}^{t}RC(min)$ $^{t}ADL = 8 * {}^{t}WC(min)$ $^{t}CCS = 8 * {}^{t}WC(min)$ $^{t}RHW = 8 * {}^{t}WC(min)$	$^{t}CK = {}^{t}CK(avg)$ $^{t}ADL = 16 * {}^{t}CK(avg)$ $^{t}CCS = 32 * {}^{t}CK(avg)$ $^{t}RHW = 16 * {}^{t}CK(avg)$
Bus idle data pattern	IO[7:0] = FFh IO[15:0] = FFFFh	DQ[7:0] = FFh
Repeated data pattern (Used for Icc2 and Icc4w)	IO[7:0] = A5h, AAh, 5Ah, 55h IO[15:0] = A5A5h, AAAAh, 5A5Ah, 5555h	DQ[7:0] = A5h, AAh, 5Ah, 55h
Array preconditioning for Icc4r	The array is preconditioned to match the following repeating data pattern: IO[7:0] = A5h IO[15:8] = A5A5h	The array is preconditioned to match the following repeating data pattern: DQ[7:0] = A5h

Test Sequences: Array

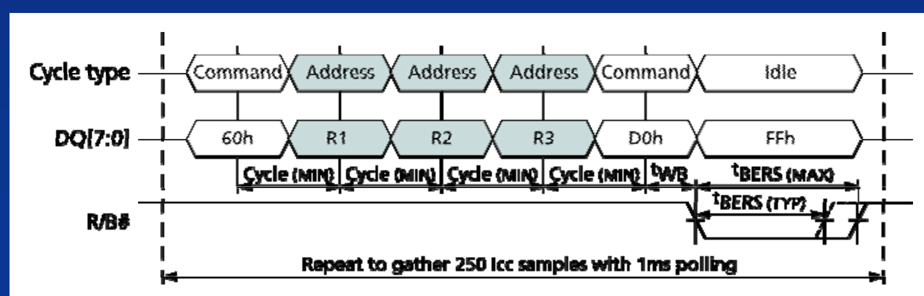
- Icc1 (Array read)



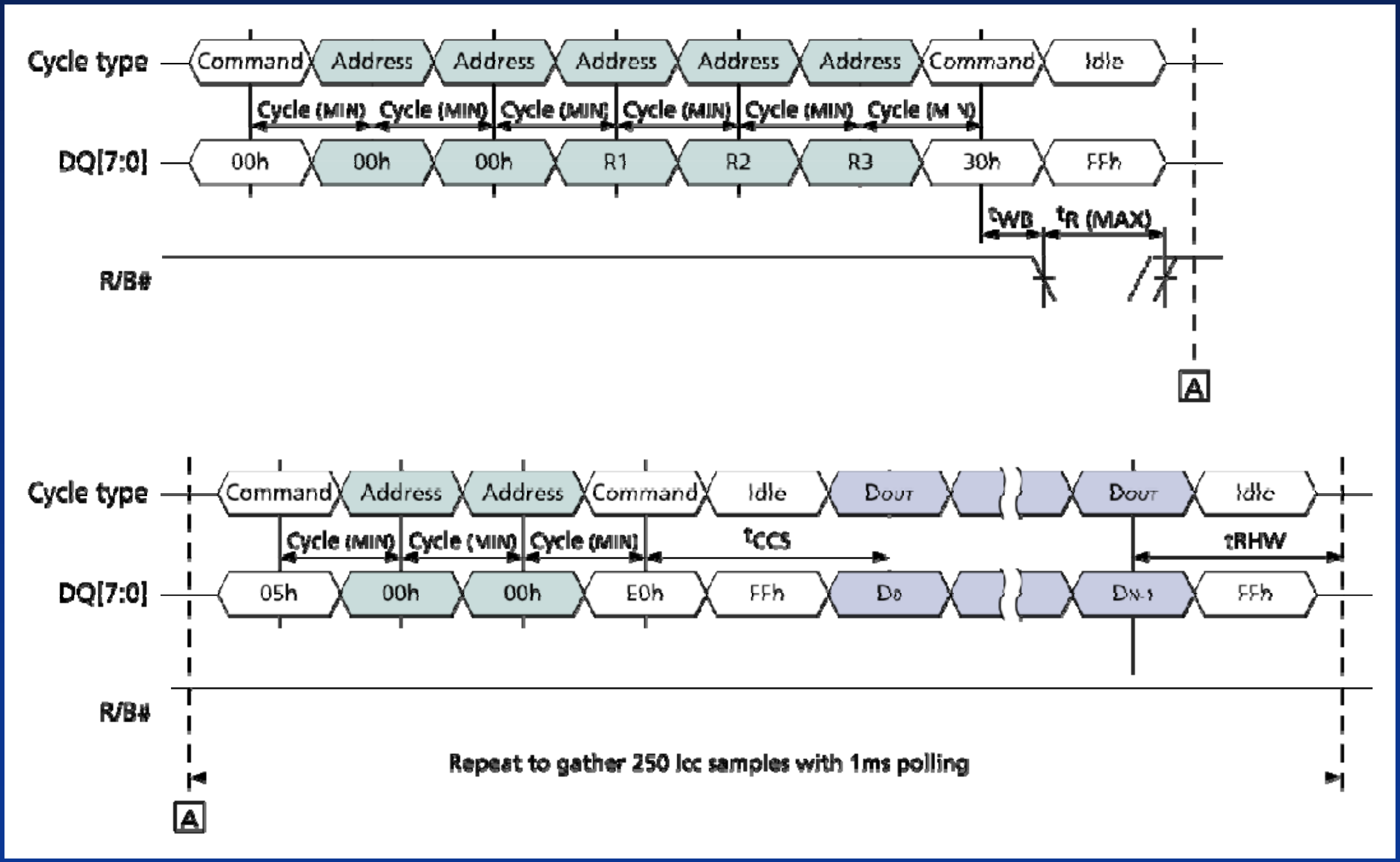
- Icc2 (Array program)



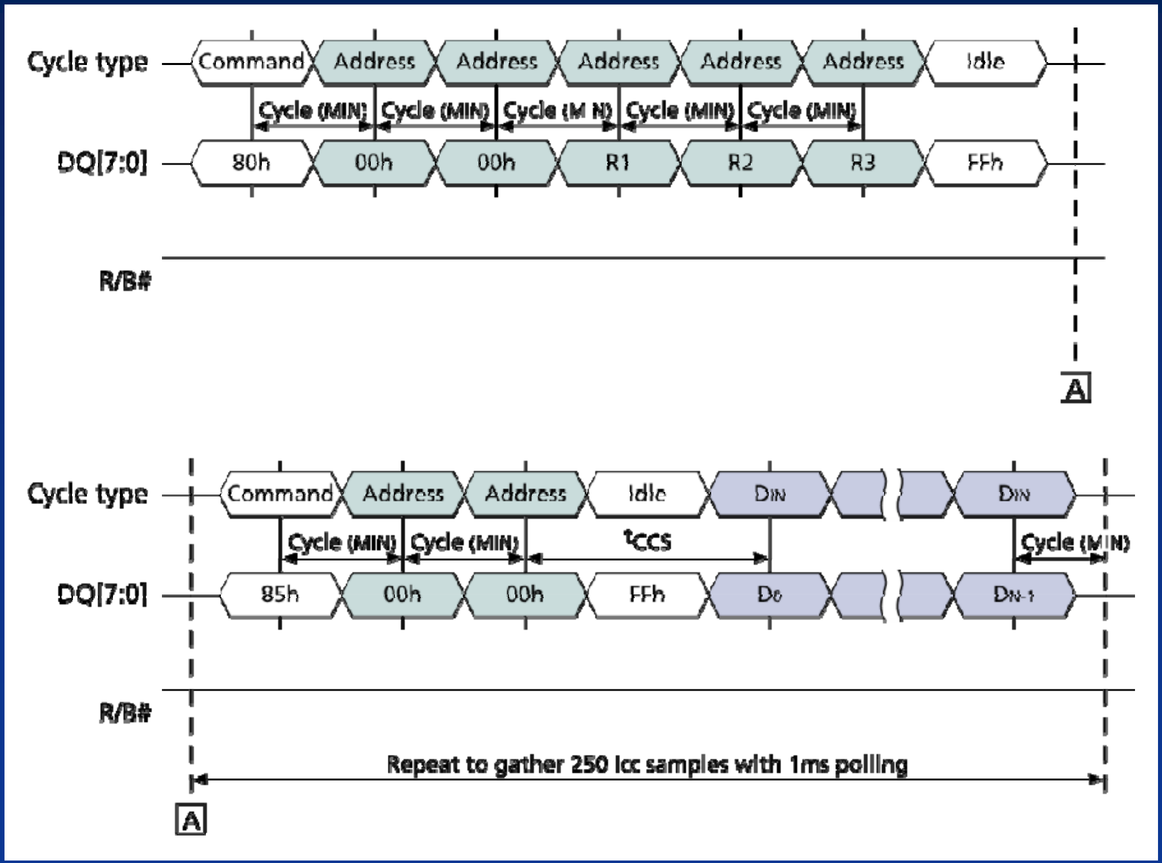
- Icc3 (Array erase)



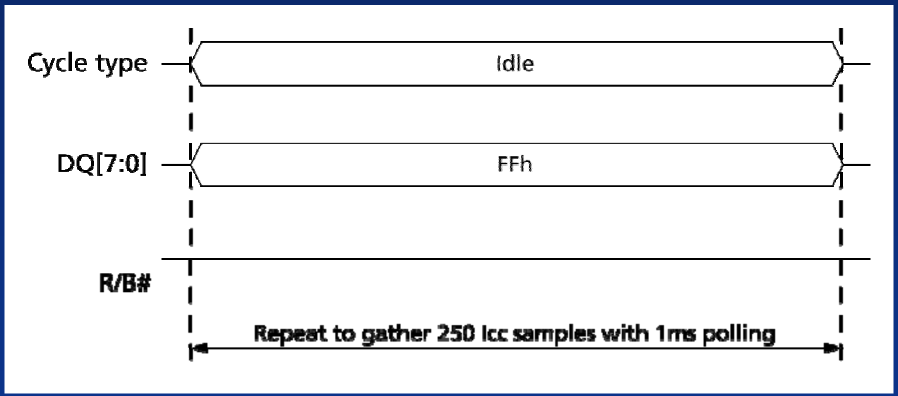
Test Sequence: Icc4r (I/O Burst Read)



Test Sequence: Icc4w (I/O Burst Write)

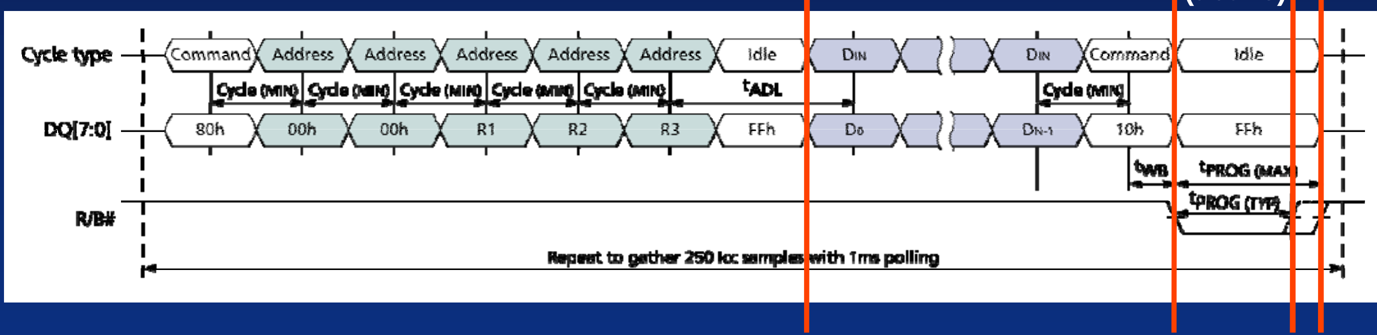


Test Sequence: Icc5 (Bus Idle)



Formulas for Correlation

- Icc2 (measured) includes Icc4w, Icc2(active), and Icc5



- Icc2 (measured) can be represented as:

$$Icc2(measured) = \frac{tIO}{tIO + tPROG(max)} Icc4w + \frac{tPROG(typ)}{tIO + tPROG(max)} Icc2(active) + \frac{tPROG(max) - tPROG(typ)}{tIO + tPROG(max)} Icc5$$

$$tIO = NAND\ Page\ Size(bytes\ (x8) or\ words\ (x16)) \times tWC(min)$$

What is the Active Icc2 Current During tPROG?

- Solve for Icc2(active)

$$Icc2(active) = \frac{Icc2(measured) \times (tIO + tPROG(max)]}{tPROG(typ)} - \frac{tIO \times Icc4w}{tPROG(typ)} - \frac{Icc5 \times tPROG(max)}{tPROG(typ)} + Icc5$$

It is possible to solve for active currents from the measured values for Icc1, Icc2, and Icc3, which can then be used in power budget modeling

Conclusion

- This presentation shows a better approach to measuring Icc that provides better predictability in the system of how much current devices will draw based on application-specific usage models
- This test methodology is
  - HVM capable
  - Reproducible
  - Usable for application-specific power budgets
  - Adaptable to multiple NAND interfaces

Questions and comments?

- This presentation does not include every detail of the new test methodology
- Micron welcomes feedback and suggestions for improvement; please contact the presentation author
- The test methodology is subject to further change and improvement

About Michael Abraham

- Manager of Micron's NAND Flash Applications Engineering group
- B.S. in Computer Engineering from Brigham Young University
- Technical representative for Micron in ONFI and JEDEC for NAND Flash
- Key role in defining and standardizing the high-speed, synchronous DDR NAND interface within Micron and at ONFI