Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

THIS SPEC IS OBSOLETE

Spec No: 001-47864

Spec Title: INTERFACING TI OMAPV1030 PROCESSOR TO ASTORIA'S PSEUDO-NAND PROCESSOR-PORT - AN47864

Sunset Owner: Dhanraj Rajput (DBIR)

Replaced by: None

Interfacing TI OMAPV1030 Processor to Astoria's Pseudo-NAND Processor-port

AN47864

Author: Hingkwan Huen
Associated Project: No
Associated Part Family: CYWB0224ABx
Software Version: Astoria™ SDK Version 1.2.1
Associated Application Notes: AN46712, AN46860

Application Note Abstract

Cypress West Bridge® Astoria™ provides high speed USB peripheral and mass storage control capabilities to the system processor through its host processor port. AN46712 presents a physical interconnect example of interfacing the TI OMAPV1030 baseband processor to the Astoria device, using Astoria's Pseudo-NAND Processor-port interface.

Introduction

The rapid growth of the mobile and embedded device market demands high speed USB connectivity and mass storage control capabilities. The Cypress West Bridge® Astoria™ device (CYWB0224ABx) is designed to enable handset designers to easily add these functionalities to their designs.

When Astoria is integrated into a system, its host Processor-port (P-port) interface is used to connect to the system bus of the principal processor. P-port on the Astoria supports multiple configurations that are compatible with a vast majority of interfaces available in many commonly used embedded processors. The Pseudo-NAND (PNAND) P-port interface is a unique feature of the Astoria. It enables the Astoria to emulate a NAND flash device. Astoria's PNAND boot feature allows a system processor to communicate with the Astoria using standard NAND commands, and boot directly from a NAND flash connected to the Astoria. The PNAND boot feature enables robust system design by consolidating boot and mass storage data in a single SLC NAND device. This feature reduces BOM cost by removing the need for having two NAND devices on the board serving different purposes: one for booting the processor and the other for mass storage.

Astoria also has the ability to allow processors to boot indirectly from SD or USB through the PNAND interface for processors that do not support the SD boot option.

Using the OMAPV1030 processor as an example, this application note describes how a system processor can interface to the Astoria through the PNAND interface, and explains the considerations involved. In particular, the physical interconnect and timing considerations between OMAPV1030 and the Astoria are explained in detail. For further details of Astoria PNAND interface, supported commands, and its mode of operations, see the Astoria datasheet and the application note AN46712.

West Bridge® Astoria

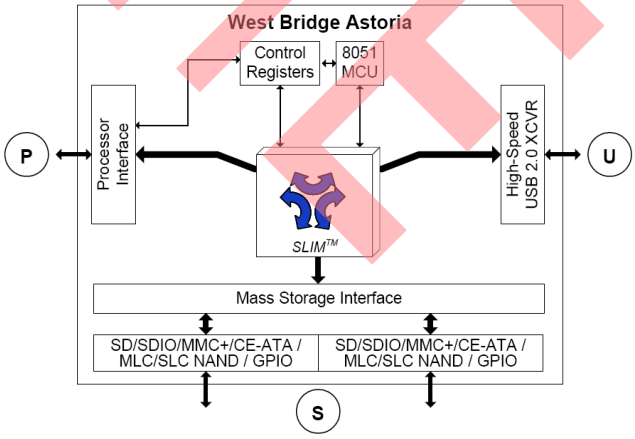
Astoria is a flexible bridge device that uses robust SLIM™ architecture that allows simultaneous and independent data paths between the processor and USB, USB and mass storage, and processor and mass storage.

The Processor-port (P-port) of the Astoria can operate in one of the multiple selectable interface types such as the CRAM, SRAM, Address-Data Multiplexed RAM, SPI, or PNAND.

The USB-port (U-port) is a High-Speed (HS) USB peripheral that is capable of independent U⇔P and U⇔S data transfers with optimal performance.

The mass storage-port (S-port) can be configured to simultaneously interface with multiple mass-storage devices such as 8- or 16-bit SLC NAND, Managed NAND, SD/MMC/SDIO, and CE-ATA devices.

Figure 1. West Bridge Astoria Block Diagram



For a full description of Astoria's features, see the West Bridge Astoria datasheet.

Astoria's PNAND Interface

Astoria's P-port NAND interface is called Pseudo-NAND (PNAND) interface. Table 1 shows the PNAND signals that are mapped to P-port of the Astoria.

There are two modes of operation in the PNAND interface:

- Logical NAND Access
- NonLogical NAND Access

Logical NAND Access (LNA) refers to the mode of operation where the Astoria emulates a NAND flash device. This mode is designed for systems that require booting of the system processor from a NAND flash device. In this type of application, the system processor can communicate with the Astoria using common NAND commands to boot from a NAND flash connected to Astoria's S-port. In this mode of operation, the Astoria mimics a real NAND device and allows the system processor to use its internal boot-ROM to boot from the Astoria, exactly as it would boot from a NAND flash. In LNA mode, the Astoria must be initialized with its attached NAND device before the system processor can access it.

In nonLNA mode of operation, the system processor interfaces to the Astoria using standard NAND interface, but does not use standard NAND commands. In nonLNA mode, the Astoria responds to a subset of NAND commands. In this mode of operation, the system processor uses a set of APIs provided by Cypress to communicate through its NAND controller to the Astoria. The nonLNA mode of operation can be used independent of the PNAND LNA boot.

When Astoria is used in PNAND LNA mode, its available features are limited to the transfers between P-port and U-port, and NAND flash access on the S-port. To take advantage of Astoria's features, the system must switch from LNA to nonLNA mode of operation. This is done by setting a register after booting is complete. The register setting is handled through the APIs. Although P⇔U access is available in LNA mode, Astoria APIs must be used to access this feature.

The PNAND interface of Astoria can be configured to support one of the following modes:

- 8-bit small block device (SBD)
- 8-bit large block device (LBD)
- 16-bit SBD
- 16-bit LBD

In SBD mode, there is one address cycle for the column address and three address cycles for the row address whereas in LBD mode, there are two address cycles for the column address and three address cycles for the row address. Moreover, in SBD mode, the acceptable commands are 00h, 01h, or 50h whereas in LBD mode, the only command is 00h.

SBD or LBD mode is configured through the external pin settings shown in Table 2. The 8 versus 16-bit mode of operation is configured by changing the Astoria's internal registers.

Note
1. The OMAPV1030 NFC interface only supports 8-bit data IO, IO[15:8] are do not cares and must be tied to a known state.

Table 2

Table 1. Astoria PNAND Interface Signals and Example Connection to OMAPV1030's NFC Interface

P-port Signal Name	IO	Alias PNAND Signal Name	OMAPV1030 NFC Interface Pins	Function	Comments
CE#	I	CE#	nfls_ce1 or nfls_ce2	Chip enable	
A[0]	I	CLE	nfls_cle	Command latch enable	
A[1]	O	R/B#	nfls_rdy	Ready/Busy#	This signal is NOT open-drained
A[3:2]	I	A[3:2]	-	PNAND mode select	A[3:2]=0b00
A[4]	I	WP#	-	Write protect	Optional
A[5]	O	SCL	-	I^2C serial clock	Used for EEPROM boot. Pull-up needed
A[6]	IO	SDA	-	I^2C serial data	Used for EEPROM boot. Pull-up needed
A[7]	I	SDBnLBD	-	SDB or LBD select	SDB if A[7]=1; LBD if A[7]=0
DQ[15:0]	IO	IO[15:0]	nfls_fd[7:0]	Data bus	Supports 8-bit or 16-bit mode
ADV#	I	ALE	nfls_ale	Address latch enable	
OE#	I	RE#	nfls_re	Read enable	
WE#	I	WE#	nfls_we	Write enable	
INT#	O	INT#	GPIOx	Astoria device interrupt	Used only in nonLNA mode
DRQ#	O	DRQ#	-	DMA request	Optional. Used only in nonLNA mode
TEST[2:0]	I	TEST[2:0]	-	Mode configuration	TEST[2:0]=0b010 for PNAND

Note
1. The OMAPV1030 NFC interface only supports 8-bit data IO, IO[15:8] are do not cares and must be tied to a known state.

Table 2. PNAND Mode Configuration Pin Setting

Mode	TEST[2]	TEST[1]	TEST[0]	A[7]	A[3]	A[2]
SBD	0	1	0	1	0	0
LBD	0	1	0	0	0	0

Interface to OMAPV1030 through PNAND

The OMAPV1030 E-GPRS baseband processor belongs to the Texas Instruments OMAP-Vox processor family. It combines a communication engine, an application engine, and a digital signal processing (DSP) subsystem. The DSP subsystem delivers enhanced multimedia performance and communication capability for the handset designers who want to develop a cost-efficient GSM/GPRS/EDGE solution for mainstream wireless users.

Among OMAPV1030's memory interfaces, the external memory interface slow (EMIFS) and NAND flash controller (NFC) interface are both compatible with the Astoria's PNAND interface.

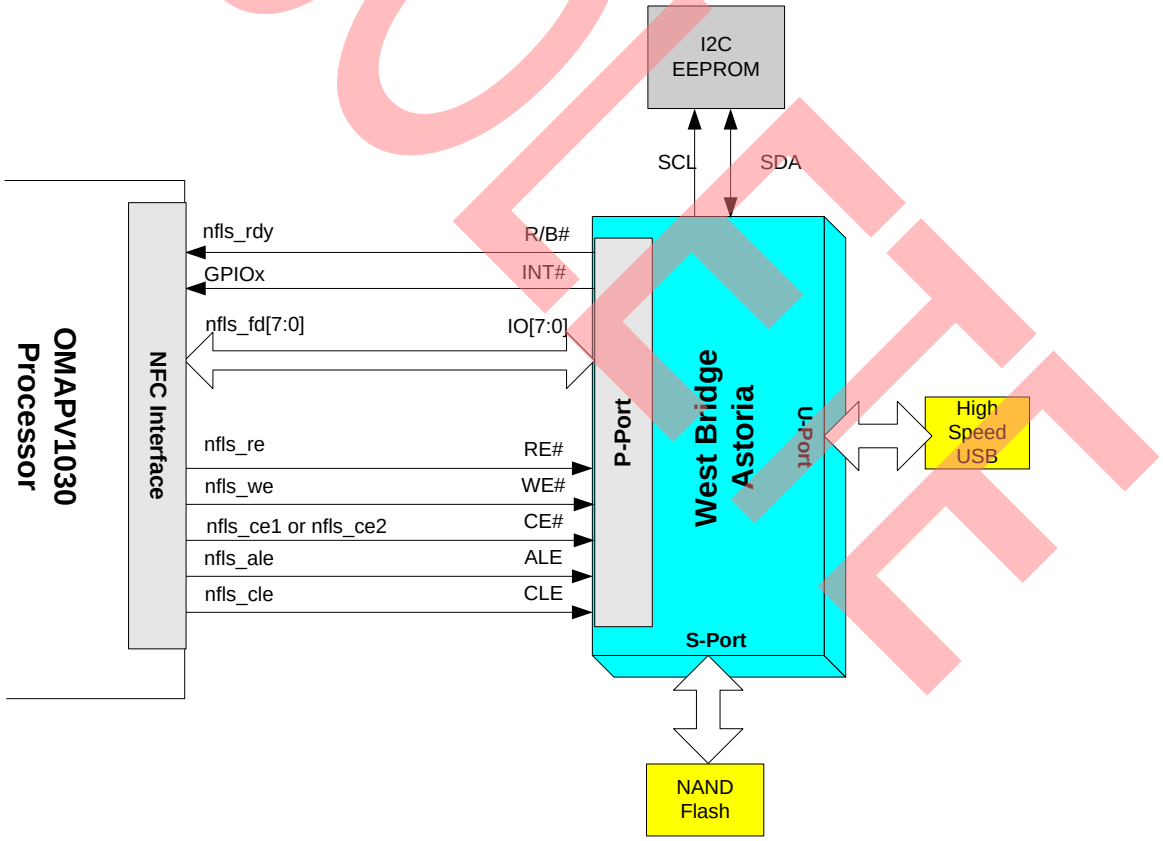
OMAPV1030's EMIFS interface can emulate a NAND flash interface through software. In this case, the Figure 2 shows an example of a connection between the OMAPV1030 processor and the Astoria.

processor supports both 8-bit and 16-bit NAND devices. However, one drawback is the additional processing power required for the emulation.

By using OMAPV1030's embedded NFC interface, although it supports only 8-bit NAND devices, a simpler setup and gain in MIPS consumption is offered, comparing to the EMIFS NAND emulation option. The example presented in this application note uses the NFC interface to interconnect OMAPV1030 with the Astoria. This is because it is easier to configure with the additional benefit of using the hardware NAND controller of the processor.

The OMAPV1030's embedded NFC interface allows seamless connection to the Astoria's P-port in PNAND configuration.

Figure 2. Example of a Connection between OMAPV1030's NFC Interface and Astoria's P-Port in PNAND Mode



The basic modes of operations of the Astoria PNAND interface are the same as common NAND devices, which include:

- **Bus Operations**

The bus on the Astoria PNAND is internally multiplexed. Data IO, addresses, and commands share the same IO pins. IO[15:8] are used only for data in the x16 configuration. Addresses and commands are always supplied on IO[7:0]. Because the NFC interface of the OMAPV1030 only supports 8-bit data IO, IO[15:8] are do not cares and must be tied to a known state. The command sequence normally consists of a command latch cycle, an address latch cycle, and a data cycle - either read or write.

- **Control Signals**

The signals CE#, WE#, RE#, CLE, and ALE control flash device READ and WRITE operations. CE# is used to enable the device. When CE# is low and the device is not in the busy state, the flash memory accepts command, data, and address information. When the device is not performing an operation, the CE# pin is typically driven HIGH and the Astoria is inactive.

- **Commands (ALE=0, CLE=1)**

All the NAND operations (except READ STATUS and RESET commands) consist of a command write cycle followed by an address write cycle. The READ STATUS command does not have an address write cycle. The command is transferred into the NAND command register followed by the start address for the read or program operation, latched into the address register. Commands are written to the command register on the rising edge of WE# when CE# and ALE are low, and CLE is high. Commands are input on IO[7:0].

- **Address (ALE = 1, CLE = 0)**

Addresses are written to the address register on the rising edge of WE# when CE# and CLE are low, and ALE is high. Addresses are input on IO[7:0].

- **Data (ALE = 0, CLE = 0)**

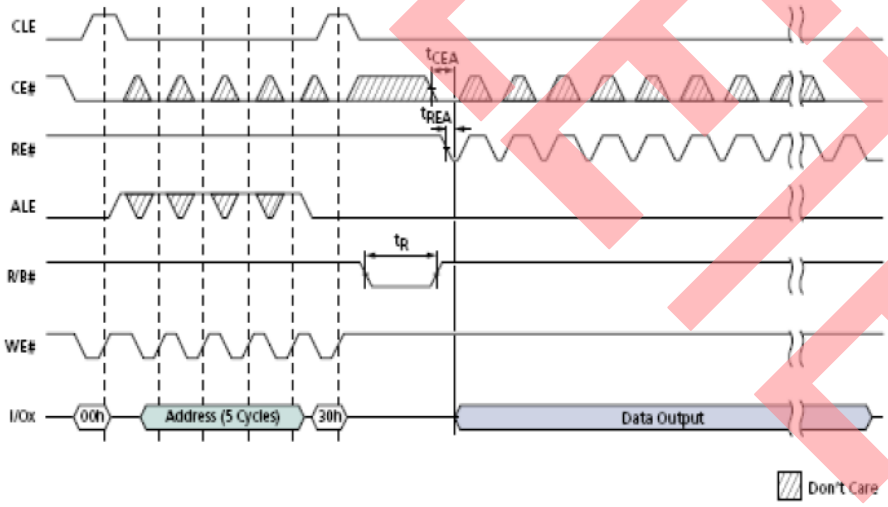
Data is written to the data register on the rising edge of WE# when CE#, CLE, and ALE are low. Data is input on IO[7:0].

- **Ready / Busy**

The R/B# output from the Astoria provides a hardware method of indicating the completion of a PROGRAM/ERASE/READ operation. The signal is typically high, and transitions to low after the appropriate command is written to the device. Because R/B# signal in the Astoria is not open-drained, there is no need to include a pull-up resistor for proper operation of this pin.

Figure 3 shows the example waveform of the successive accesses: COMMAND Latch, ADDRESS Latch, and DATA Output. Note that the commands cannot be sent to the NAND flash during tR, because it is busy.

Figure 3. Astoria PNAND Page Read Operation



See the Astoria's datasheet and AN46712 for supported command sets and full operation description of PNAND interface.

OMAPV1030 NFC Interface Configuration

Clocking

The OMAPV1030 NFC module is clocked by a functional clock (NDFLASH_FLCK) and an interface clock (NDFLASH_ICLK). Both clocks are driven by the processor's MPU peripheral clock (ARMPER_CK) derived from the OMAP3.4 clock generator one module (CLKM1). The ARMPER_CK clock frequency can be programmed through the ARM_CLKCTL[0:1] register mapped into the OMAPV1030 MPU address space. Because each system may have a different setting for the ARMPER_CK, this application note assumes that:

ARMPER_CK frequency =130 MHz

That is, ARMPER_CK period = 7.7 ns

If ARMPER_CK runs at a different frequency, the interface timing must be re-evaluated, but the principle of the analysis remains the same.

Timing Registers

To correctly interface the OMAPV1030 processor to the Astoria, the OMAPV1030 NFC registers must be configured. Table 3 shows the list of NFC timing registers that need to be configured.

Table 3. OMAPV1030 NFC Interface Timing Registers

Register Name	MPU Mapped Address	Description
NND_PSC_CLK	0xFFFBCC5C	A 4-bit value in this register divides the interface clock to adjust the timing for signals on the physical interface
NND_PSC1_CLK	0xFFFBCC8C	NAND controller clock prescale register HIGH
NND_PSC2_CLK	0xFFFBCC90	NAND controller clock prescale register SAMPLE

The NND_PSC1_CLK and NND_PSC2_CLK registers are used with the NND_PSC_CLK register to control the memory interface timing. To maintain backward compatibility with other OMAP devices (for example, OMAP1610), any value written to the NND_PSC_CLK register is copied into both NND_PSC1_CLK and NND_PSC2_CLK. As a result, NND_PSC1_CLK and NND_PSC1_CLK must be written after the NND_PSC_CLK.

Timing Analysis and Recommendations

This section provides a timing analysis summary between the OMAPV1030 and the Astoria.

Table 4 on page 5 shows the OMAPV1030 NFC's interface timing specifications. The specification values are converted in terms of timing register values for easier calculation of the recommended final program values.

An accurate one to one comparison is necessary between the OMAPV1030 NFC interface and the Astoria PNAND waveforms. Figure 4 and Figure 5 on page 5 show two cases that highlight all the required timings.

Table 5 on page 6 presents compatible timing parameters and calculations for the required register configurations. Based on these calculations, the recommended OMAPV1030 NFC interface register settings are:

- X = NND_PSC_CLK register value = 0x3;
- Y = NND_PSC1_CLK register value = 0x1;
- Z = NND_PSC2_CLK register value = 0x0.

To determine the throughput and optimization of the interface with a particular processor, contact your local Cypress sales representative.

Table 4. OMAPV1030 NFC Interface Timing Specifications

Symbol	OMAPV1030 Parameter	Description	OMAPV1030 Spec		OMAPV1030 Spec in Terms of Register Value	
			Min (ns)	Max (ns)	Min (ns)	Max (ns)
NFC1	tW(WEV)	nfls_we low duration	P-1.5	P+1.5	(X+1)*7.7-1.5	(X+1)*7.7+1.5
NFC2	tW(WEIV)	nfls_we high duration	Q-1.5	Q+1.5	(Y+1)*7.7-1.5	(Y+1)*7.7+1.5
NFC3	td(CLEV-WEV)	Delay time, nfls_cle high to nfls_we low	-1.9	+1.4	-1.9	+1.4
NFC4	td(WEV-IOV)	Delay time, nfls_we high to nfls_fd valid	-2.8	+0.3	-2.8	+0.3
NFC5	td(WEiV-CLEL)	Delay time, nfls_we high to nfls_cle low	P-1.4	P+1.9	(X+1)*7.7-1.4	(X+1)*7.7+1.9
NFC6	td(ALEV-WEV)	Delay time, nfls_ale high to nfls_we low	-1.9	+1.5	-1.9	+1.5
NFC7	td(WEiV-ALEL)	Delay time, nfls_we high to nfls_ale low	P-1.5	P+1.9	(X+1)*7.7-1.5	(X+1)*7.7+1.9
NFC8	tW(REV)	nfls_re low duration	P-1.7	P+1.7	(X+1)*7.7-1.7	(X+1)*7.7+1.7
NFC9	tW(REIV)	nfls_re high duration	Q-1.7	Q+1.7	(Y+1)*7.7-1.7	(Y+1)*7.7+1.7
NFC10	tsu(IOV-REH)	Setup time, nfls_fd[7:0] valud before nfls_re hgih	9.6-N	-	9.6-(Z-X)*7.7	-
NFC11	th(REH-IOV)	Hold time, nfls_fd[7:0] valud after nfls_re hgih	+4.9	-	+4.9	-

Notes

1. Let X=NND_PSC_CLK register value; Y=NND_PSC1_CLK register value; Z=NND_PSC2_CLK register value
2. P = (X + 1) * ARMPER_CK clock cycles
3. Q = (Y + 1) * ARMPER_CK clock cycles
4. N = (Z - X) * ARMPER_CK clock cycles

Figure 4. Command Latch and Address Latch Cycle (Astoria vs. OMAPV1030)

Figure 5. Serial Read Access Cycle (Astoria vs. OMAPV1030)

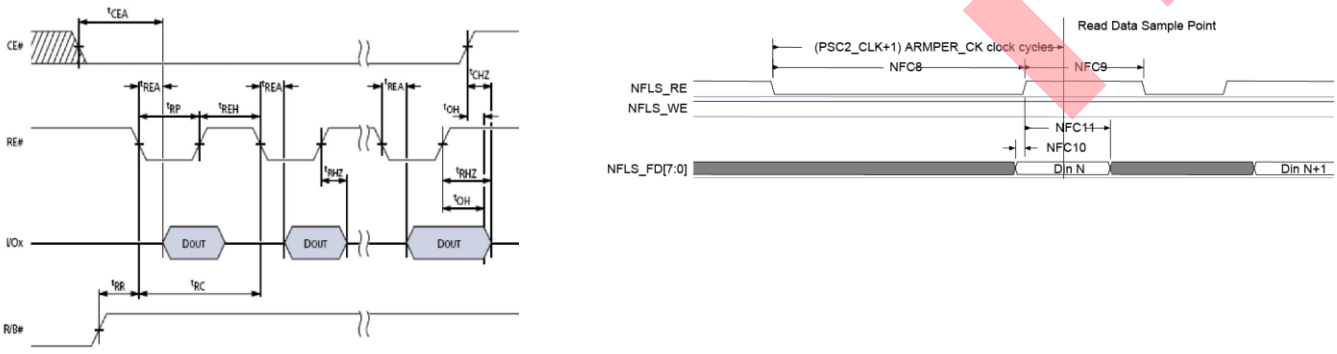


Table 5. Astoria and OMAPV1030 NFC Timing Analysis and Calculations

Timing	Astoria Name	Description	Astoria Min [ns]	Astoria Max [ns]	OMAPV1030 Parameter	Calculated NFC Configuration
Setup	tCLS	CLE Setup Time	15		td(CLEV-WEV)min + tW(WEV)min	(-1.9)+(X+1)*7.7-1.5 > 15 => X = 2
	tALS	ALE Setup Time	5		td(ALEV-WEV)min + tW(WEV)min	(-1.9)+(X+1)*7.7-1.5 > 5 => X = 1
	tCS	CE# Setup Time	20		Not programmable	Guaranteed by tCLS and tALS
	tDS	Data Setup Time	15		tW(WEV)min - td(WEV-IOV)max	(X+1)*7.7-1.5 - 0.3 > 15 => X = 0
	tRP - tREA + 2*board delay	Data Output Setup Time	-13		tsu(IOV-REH)	9.6-(Z-X)*7.7 < -13 => X = 3; Z = 0
	tCEA	CE# Access Time		35	Not programmable	
	tREA	Register/EP Access Time		30	Not programmable	
	tRR	Ready/Busy# to RE LOW	20		Managed by software	
Hold	tCLH	CLE Hold Time	5		td(WEiV-CLEL)min	(X+1)*7.7-1.4 > 5 => X = 0
	tALH	ALE Hold Time	5		td(WEiV-ALEL)min	(X+1)*7.7-1.5 > 5 => X = 1
	tCH	CE# Hold Time	5		Not programmable	
	tDH	Data Hold Time	5		tW(WEIV)min – td (WEV-IOV)max	(Y+1)*7.7-1.5 - 0.3 > 5 => Y = 0
	tOH	Data Output Hold Time	15		th(REH-IOV)min	Always met
	tREH	RE# High Hold Time	10		tW(REIV)min	(Y+1)*7.7-1.7 > 10 => Y = 1
Pulse Width	tWP	WE# Pulse Width	15		tW(WEV)min	(X+1)*7.7-1.5 > 15 => X = 3
	tRP	RE# Pulse Width	15		tW(REV)min	(X+1)*7.7-1.7 > 15 => X = 3
	tWC	Write Cycle Time	30/33 (Note 1)		tW(WEV)min + tW(WEIV)min	(X+1)*7.7-1.5 + (Y+1)*7.7-1.5 > 33 => X = 3; Y = 0
	tRC	Read Cycle Time	30/33 (Note 1)		tW(REV)min + tW(REIV)min	(X+1)*7.7-1.7 + (Y+1)*7.7-1.7 > 33 => X = 3; Y = 0

Notes
1. 30 ns for Astoria VFBGA package; 33 ns for Astoria WLCSP package.
2. Assume board delay of 1 ns.

Summary

Astoria can be effectively interconnected with TI OMAPV1030 baseband processor through the Astoria's PNAND interface. The timing analysis described in this application note affirms the mode of operations supported with the Astoria. The integration of the Astoria device through a PNAND interface is advantageous for the OMAPV1030 based system designs. It provides popular USB connectivity and mass storage control capabilities, and also allows the processor to use conventional NAND booting option with the Astoria device.

The timing analysis and recommended timing numbers for the OMAPV1030 hardware NAND controller interface are also documented here.

Additional Resources

- West Bridge Astoria Advance Datasheet
- OMAPV1030 Product Bulletin
- Interfacing to West Bridge Astoria's Pseudo-NAND Processor Port – AN46712
- Schematic Review Checklist for West Bridge® Astoria™ - AN46860

About the Author

Name: Hingkwan Huen

Title: DCD Applications Engineer

Contact: hkh@cypress.com

Document History

Document Title: Interfacing TI OMAPV1030 Processor to Astoria's Pseudo-NAND Processor-port – AN47864

Document Number: 001-47864

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2546798	HKH/AESA	08/04/2008	New application note.
*A	3185716	ANOP	03/02/2011	Updated Astoria SDK Version, Updated Introduction, West Bridge Astoria Sections and Table 1, Added Additional Resources section.
*B	3330689	ODC	07/28/2011	Updated Associated Application Notes and broken links.
*C	4108854	RSKV	08/30/2013	Obsolete application note

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com/