AARHUS SCHOOL OF ENGINEERING

MPS

Lesson 02

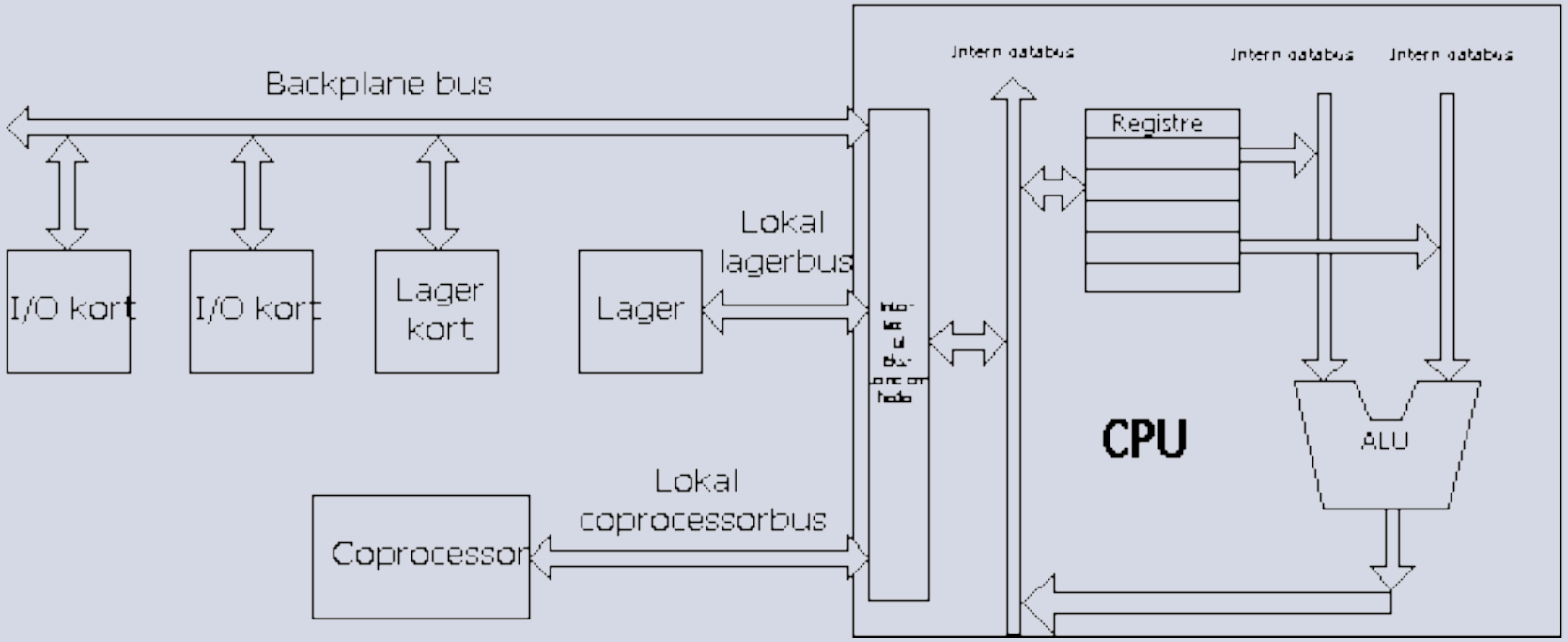
Busses

Todays Lesson

AARHUS SCHOOL OF ENGINEERING

8:00	Last Lesson
	Presentation of last exercise
	What is a bus? – An Introduction
8:40	Break (10 min)
	Signals on a Bus
	Bus Timing Diagrams
9:10	Break (10 min)
	CPU Bus Controller
	CPU Bus Hierarchy
9:30	Exercise – Estimate Bus Timing Parameters
12:00	Ahh! Lunch!

What's a Bus?


Backplane bus
I/O kort
I/O kort
Lager kort
Lager
Lokal lagerbus
Coprocessor
Lokal coprocessorbus
Intern databus
Intern databus
Intern databus
Registre
CPU
ALU

Busses in General

AARHUS SCHOOL OF ENGINEERING

- Busses are used to interconnect multiple devices
- A shared bus reduces the number of connections between devices.
- Connections on IC's (pins) are expensive!
- With one CPU, a shared bus is just as effective as separate connections
- With more, we must rely on the *Locality of reference principle*
- Different bus schemes exist:
  - Multiplexed Data / Address
  - Un-multiplexed
  - P2P "busses"

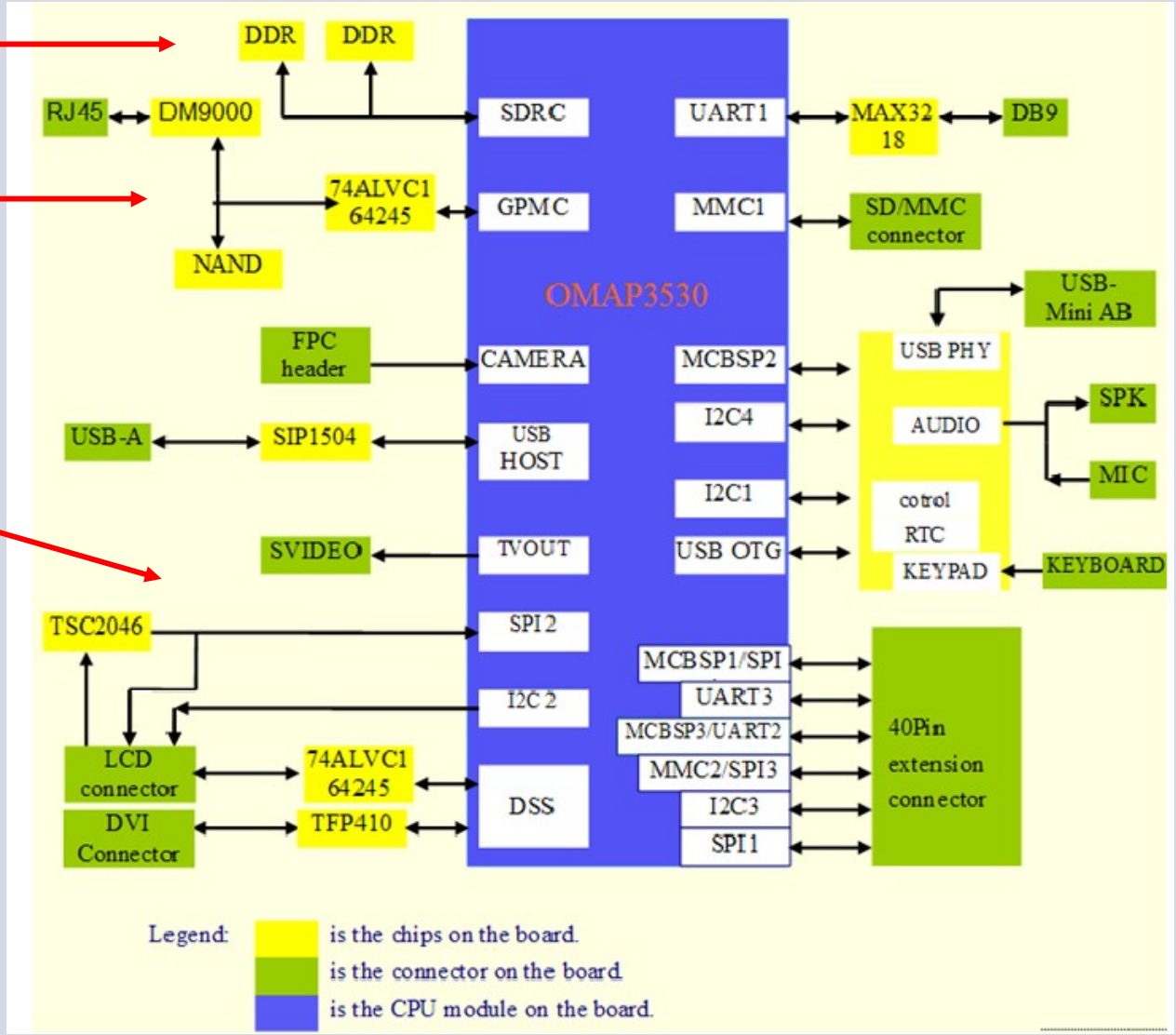
Busses on DevKit8000

AARHUS SCHOOL OF ENGINEERING

DDR RAM

GP BUS Buffered

SPI Serial Bus



On-Class Exercise

AARHUS SCHOOL OF ENGINEERING

- Open the Devkit8000 Schematics (http://devkit8000.wikispaces.com/Devkit8000Docs)
- Find answers to the following:
  - How many LEDs are on the board?
  - What does the [4] next to signal labels stand for?
  - What is “VDD18” short for?
  - What IO number does “USER_KEY” connect to?

Processor Bus Interfaces

AARHUS SCHOOL OF ENGINEERING

Select address on slave device

Data to/from slave device

Select, Read, Write and Wait signals

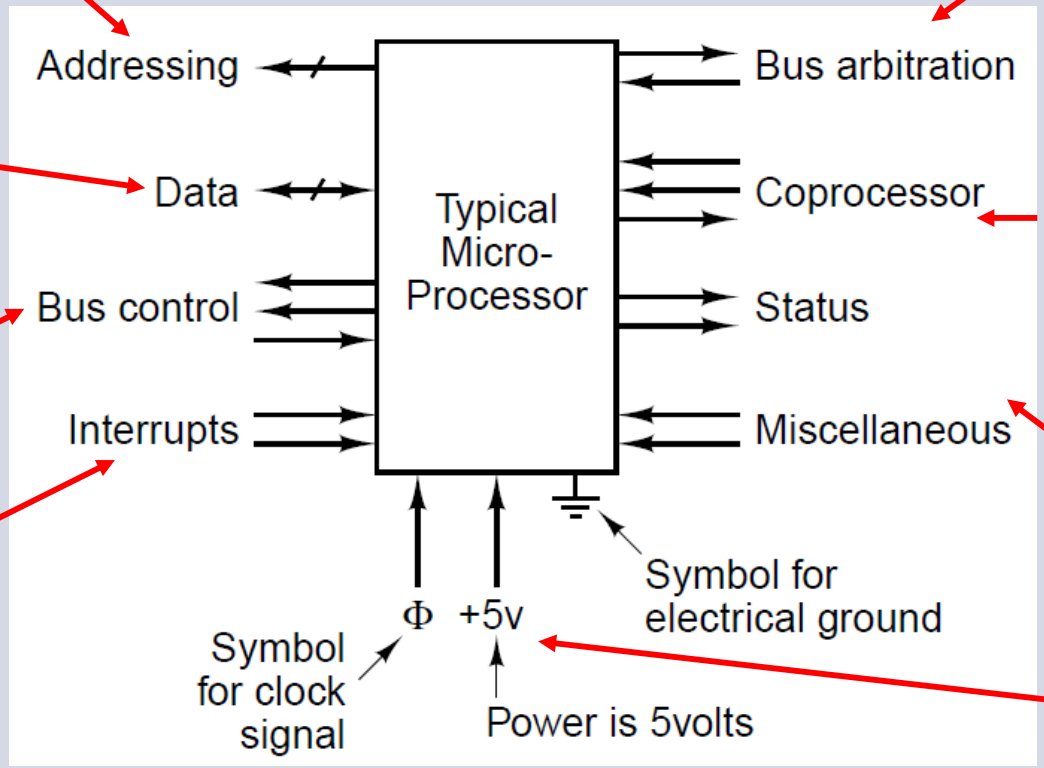
Slave interrupts

Multi-master Control signals

Special hand-shaking signals

Misc. signals

Commonly named VCC



Addressing (1:2)

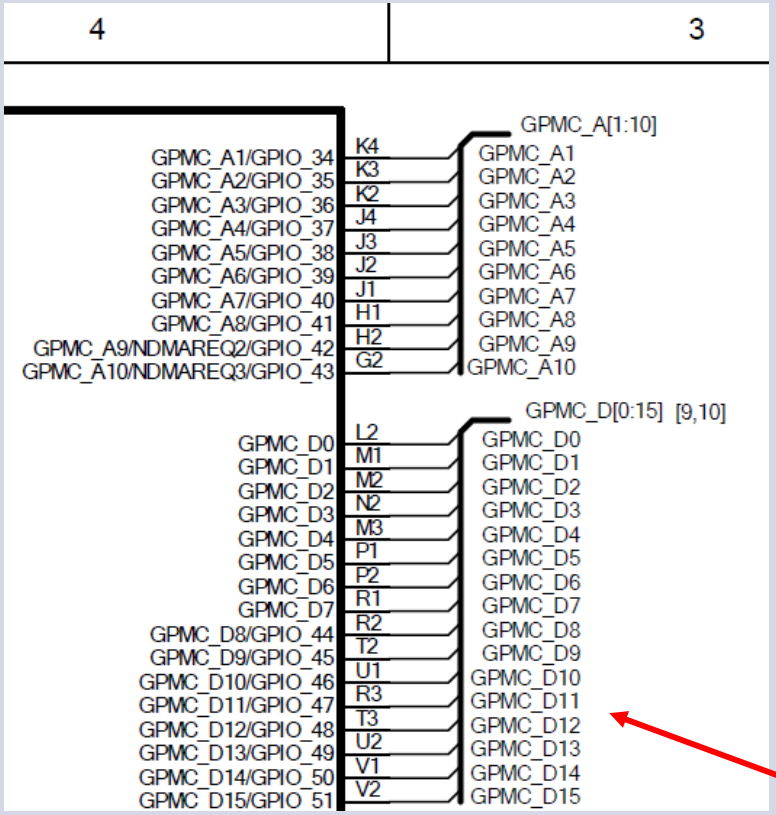
AARHUS SCHOOL OF ENGINEERING

- Number of address bits sets addressable space:
  - Size = 2^N * (Atomic size/8)
  - Atomic size = Nbr bits addressable with lowest addr bit
- Address / Data can be multiplexed:
  - Lower pin count / power
  - Lower Speed
- I/O busses typically 8,16-bit with high fanout
- SD-DDR RAM busses typically 32, 64, 128-bit w. low fanout

Addressing (2:2)

AARHUS SCHOOL OF ENGINEERING

DevKit8000 Schematic p.2

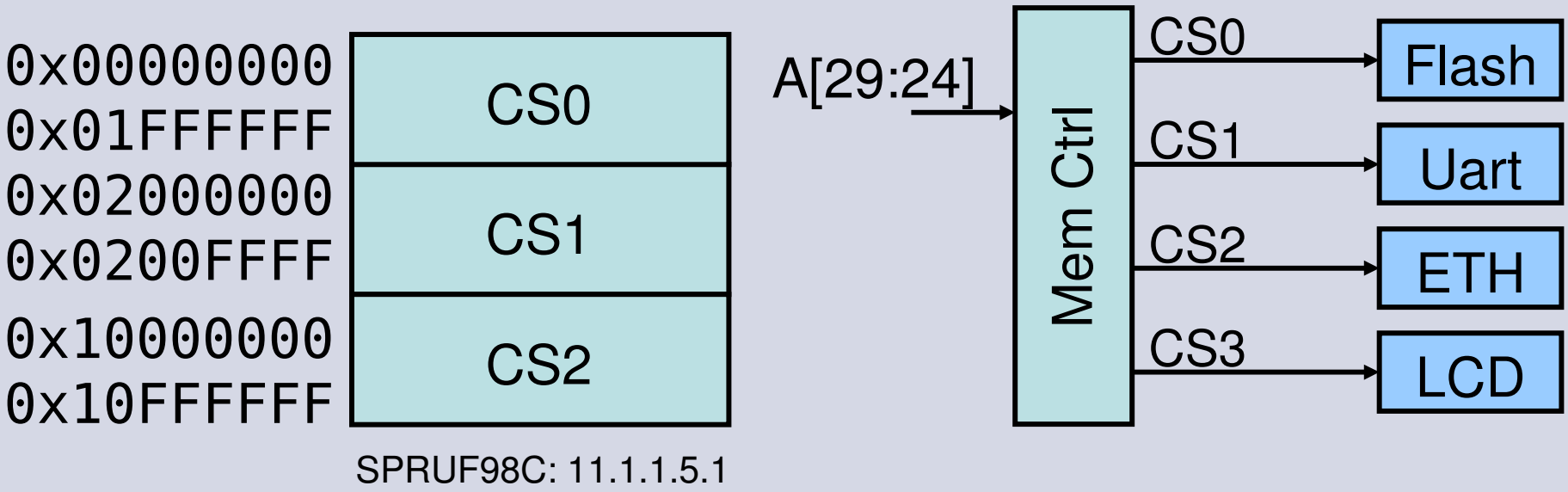


Bus Symbol

Atomic size?

Adressable Area?

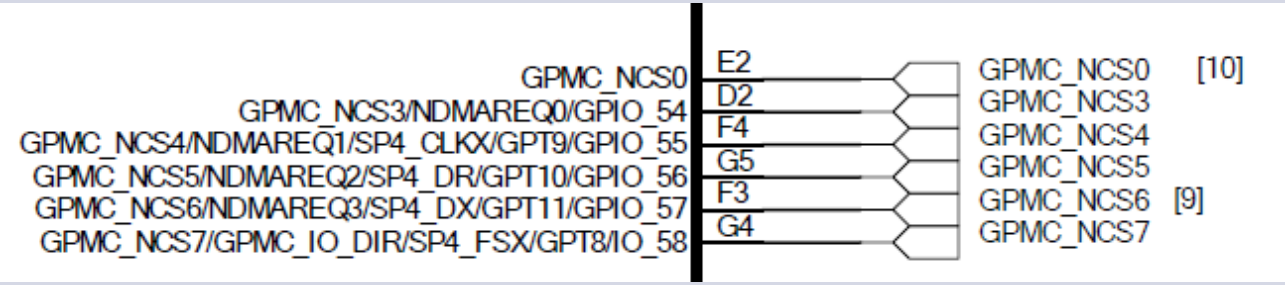
Chip Select (1:2)



- Chip Select (CS) is used to … Select Chips!
- CS is generated from decoded internal address bits
- Start address and range is often configurable

What happens if addr range is chosen bigger than device actually is?

Chip Select (2:2)

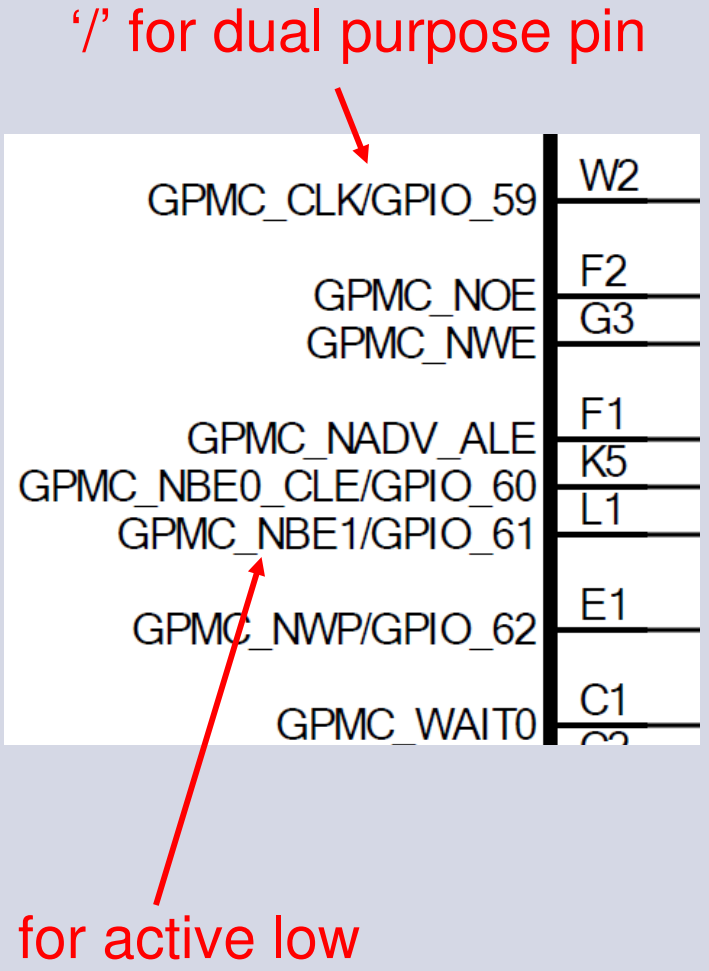


DevKit8000
Schematic p.2

Control Signals

AARHUS SCHOOL OF ENGINEERING

- RD (OE) / WR (WE)
  - Instructs device that this is a read / write access
- ALE / CLE
  - Address- / Command Latch Enable. For multiplexed busses
- BE0 / BE1
  - Byte Enable Low/High. Allows byte access on 16-bit interface
- WAIT / BUSY
  - Signal from device, requesting for time to prepare data
- CLK
  - Clock output for bus subsystem



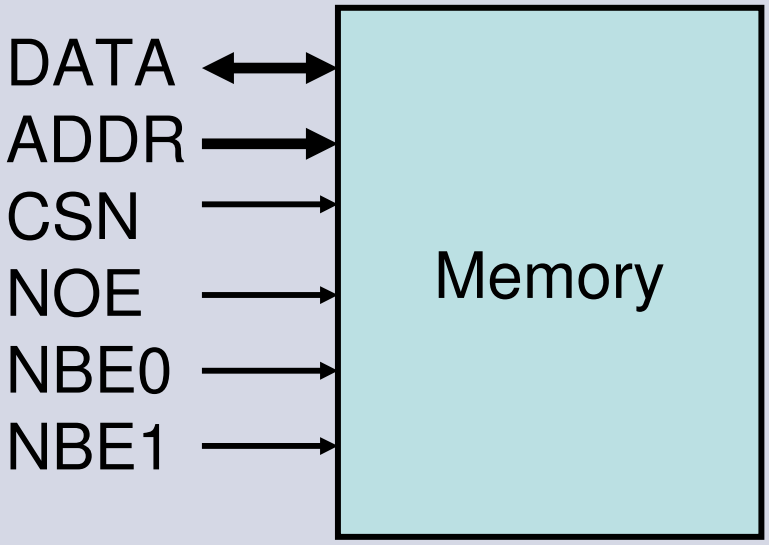
On-Class Exercise

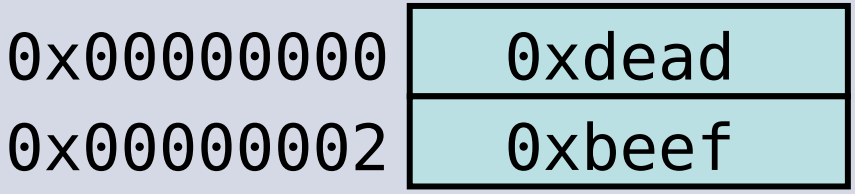
AARHUS SCHOOL OF ENGINEERING

- Open the Devkit8000 Schematics (http://devkit8000.wikispaces.com/Devkit8000Docs)
- Find answers to the following:
  - How many data bits are used for the DDR interface? (p.9)
  - What chip select is connected to the ethernet chip (DM9000)?
  - How does GPMC_D0 correlate to B_D0? (p.7)
  - What pin number on the CPU (U4) is CAM_D0 connected to? – and can that pin be used for anything else?

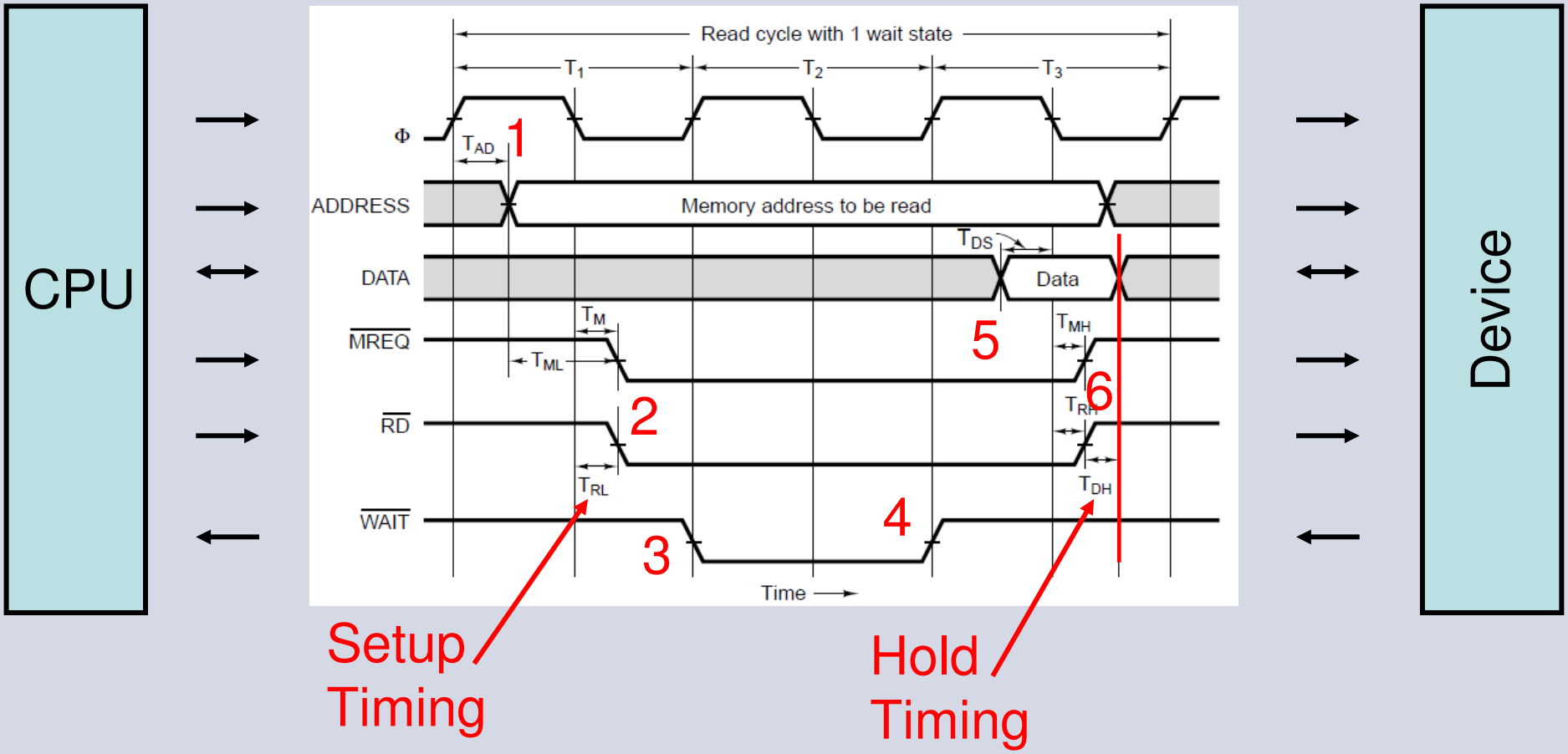
Byte Enable

- Assuming Little-Endian
- IO_READ16(0x0):
  - ADDR = 0x00000000
  - NBE0/NBE1 = 00
  - RESULT = **0xdead**
- IO_READ8(0x0):
  - ADDR = 0x00000000
  - NBE1/NBE0 = 10
  - RESULT = **0xad**
- IO_READ8(0x1):
  - ADDR = 0x00000000
  - NBE1/NBE0 = 01
  - RESULT = **0xde**





Synchronous Bus Timing

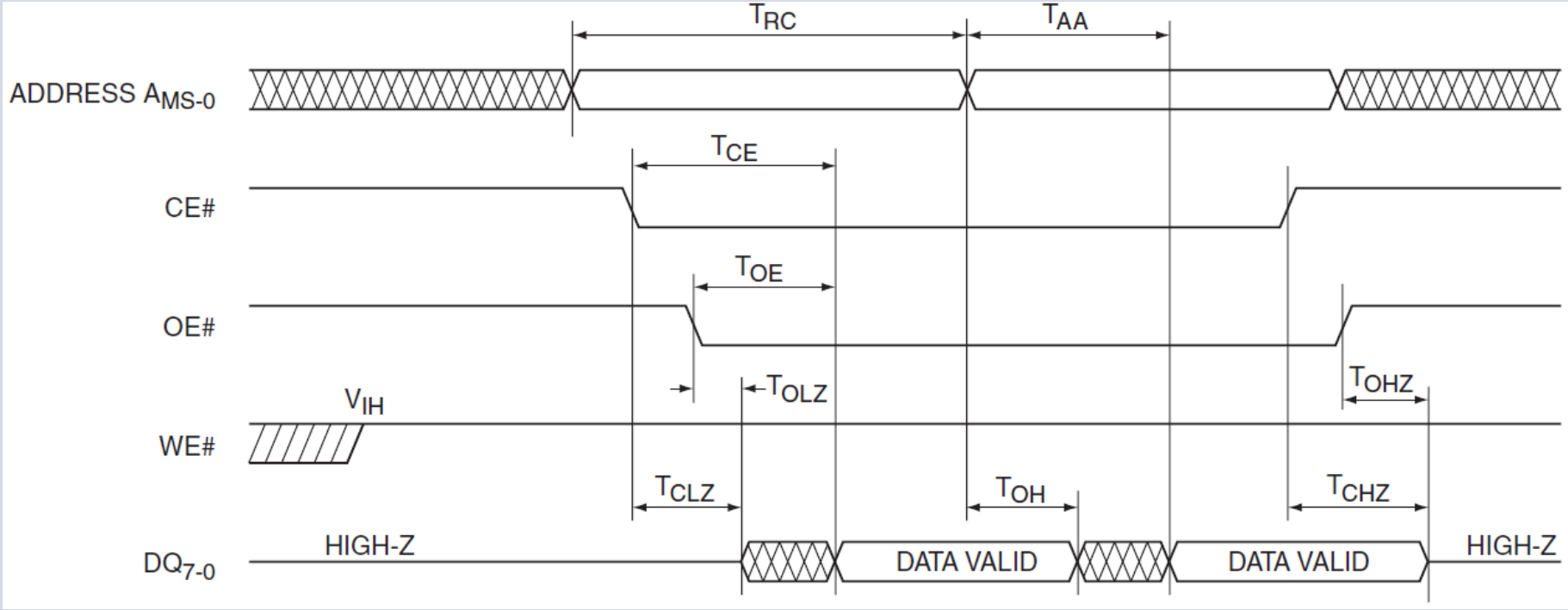


- Device asserts “WAIT” to extend timing. CPU inserts extra clock cycles (wait-states). Wait-states are given in whole clock cycles

Setup & Hold Timing

- Setup Time: Time from input data valid to sampling
- Hold Time: Time from data sampling to data input is invalid

Asynchronous Bus Timing



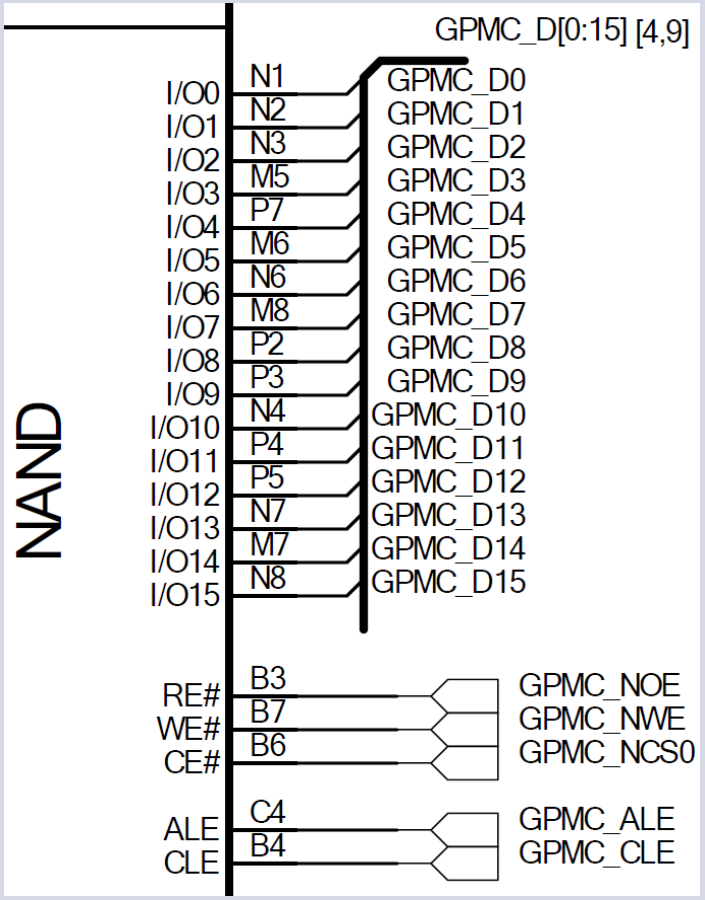
- Flash Memory and most “slow” external devices
- Fixed Timing is typically used
- “Wait” signal from slave may end cycle asynchronally
- Timing (if fixed) must be configured on bus master (CPU)

Multiplexed Example: NAND Flash

AARHUS SCHOOL OF ENGINEERING

- Address/Data Multiplexed
- Address latched with ALE signal
- Several address cycles to select address:
  - Column
  - Page
  - Bank

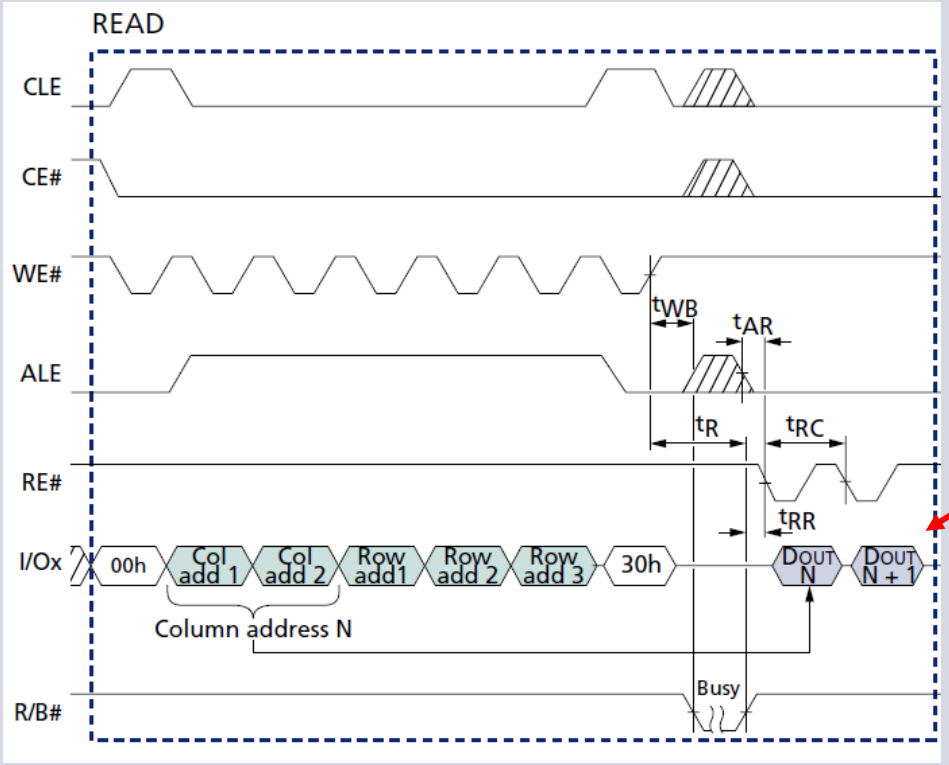
  } Row
- Huge address space must be handled by a file system
- Is not mapped directly into CPU's memory space



DevKit8000 Schematic p.8

Example: NAND Flash cont.

AARHUS SCHOOL OF ENGINEERING



Data Burst

Table 6: 2Gb SLC NAND Flash Addressing Scheme

Cycle	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
First	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
Second	LOW	LOW	LOW	LOW	CA11	CA10	CA9	CA8
Third	BA7	BA6	PA5	PA4	PA3	PA2	PA1	PA0
Fourth	BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
Fifth	LOW	LOW	LOW	LOW	LOW	LOW	LOW	BA16

Memory Controllers

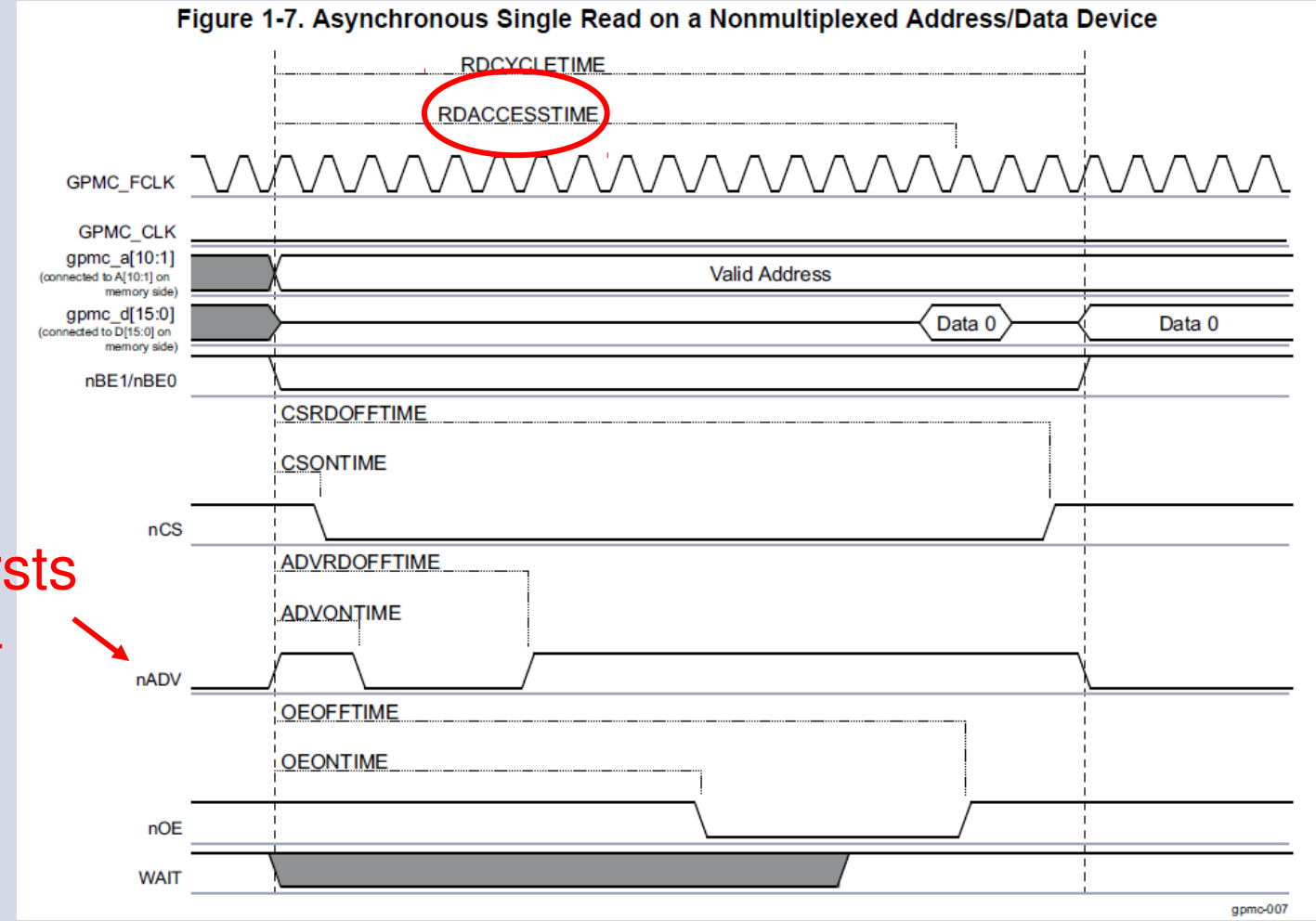
AARHUS SCHOOL OF ENGINEERING

- A memory controller adapts timing and mapping of external devices
- Usually one set of configuration parameters per chip select
- The OMAP3530 has three memory controllers:
  - GPMC – General Purpose Memory Controller
  - SDRC – SDRAM Controller
  - OMC – On-Chip Memory Controller
- The GPMC supports
  - 1 GB total address space
  - 8 Chip Selects (Programmable base address and range)
  - Multiplexed / Non-multiplexed operation
  - Burst read access
  - Little- / Big endian access

OMAP GPMC Timing Config

AARHUS SCHOOL OF ENGINEERING

Figure 1-7. Asynchronous Single Read on a Nonmultiplexed Address/Data Device



For bursts
Addr++

Spuf98c – 11.1.1.5.3

GPMC Configuration Registers

AARHUS SCHOOL OF ENGINEERING

Table 1-42. GPMC_CONFIG5_i

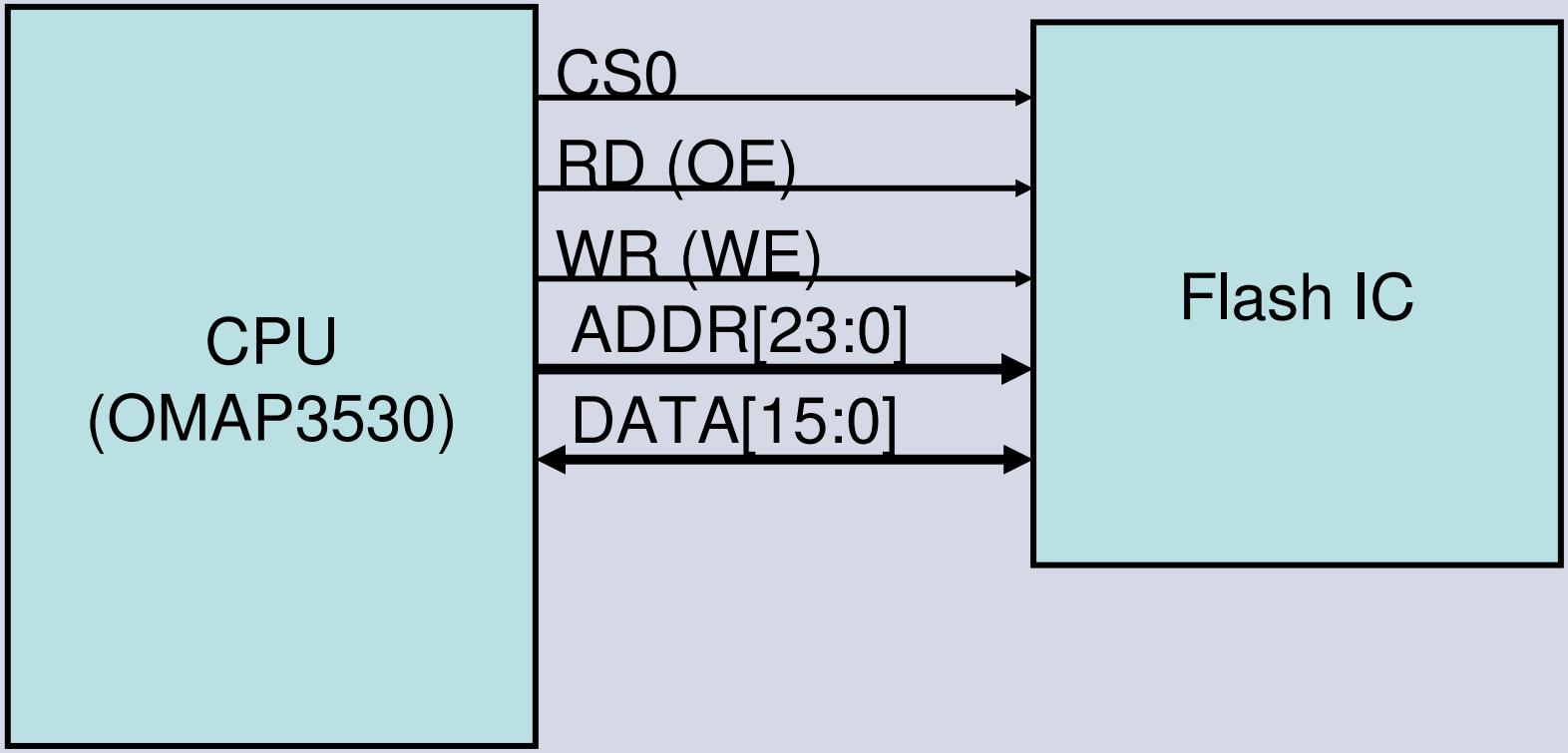
Address Offset	0x0000 0070 + (0x0000 0030 * I)	**Index**	I = 0 to 7
Physical Address	0x6E00 0070 + (0x0000 0030 * I)	**Instance**	GPMC
Description	RdAccessTime and CycleTime timing parameters configuration		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21	20 19 18 17 16	15 14 13	12 11 10 9 8	7 6 5	4 3 2 1 0
RESERVED	PAGEBURSTACCESSTIME	RESERVED	RDACCESSTIME	RESERVED	WRCYCLETIME	RESERVED	RDCYCLETIME

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
27:24	PAGEBURSTACCESSTIME	Delay between successive words in a multiple access 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x1
23:21	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
20:16	RDACCESSTIME	Delay between start cycle time and first data valid 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x0F

Spuf98c – 11.1.1.7.1

Estimating Parameter Values (1:2)

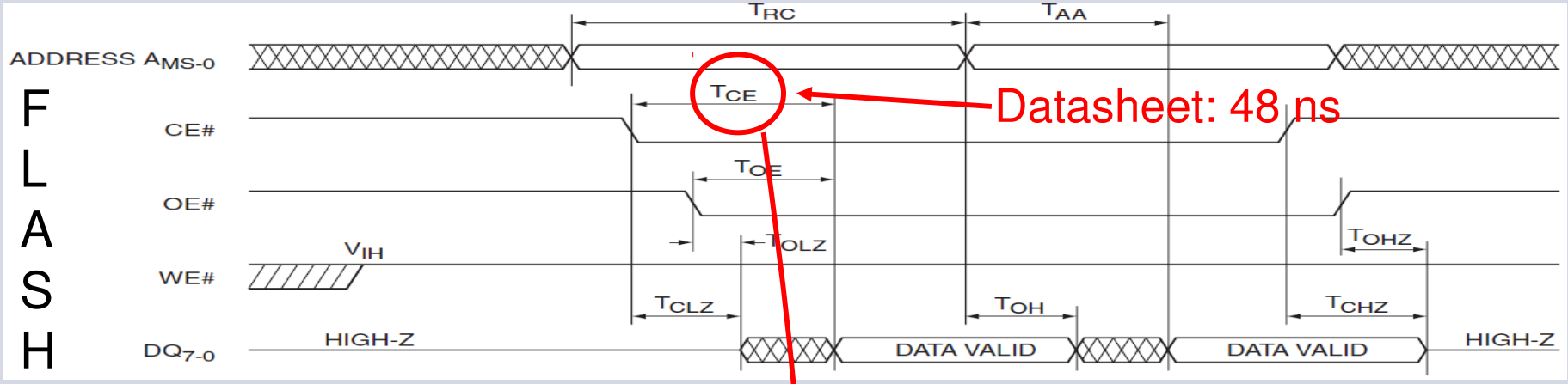


- nCS or $\overline{CS}$ indicates an active low signal
- DATA may be reffered DQ or just D

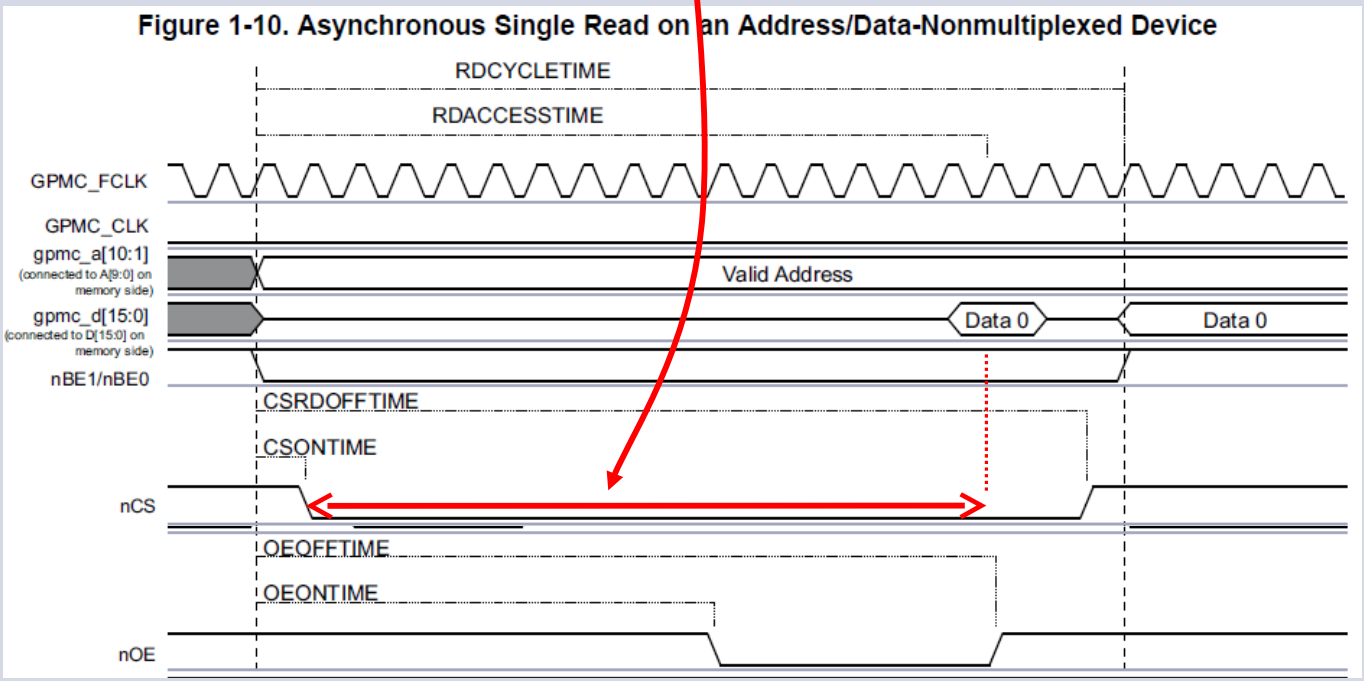
Estimating Parameter Values (2:2)

AARHUS SCHOOL OF ENGINEERING

FLASH



CPU



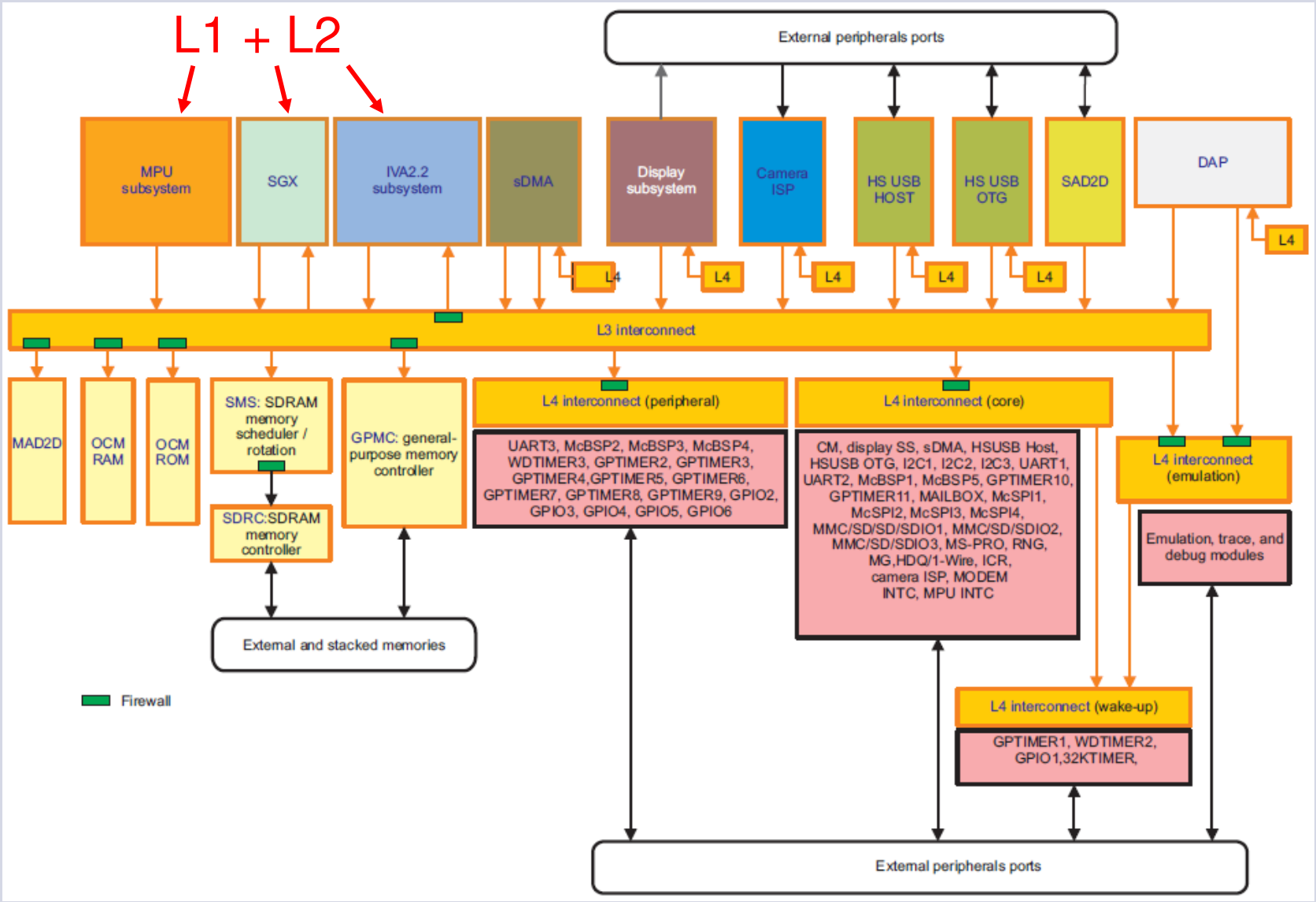
CSONTIME? RDACCESSTIME?

Busses inside the Processor

AARHUS SCHOOL OF ENGINEERING

- Several Levels of busses inside a processor
- Level 1
  - Kernel – CPU Registers – L1 Cache
  - F = Fkernel
- Level 2
  - L1 – L2 Cache interconnect
  - F = ~ Fkernel / 10
- Level 3
  - External Memory, fast peripherals, Video, PCI-Express, Gb Ethernet
  - F ~ Fkernel / 100
- Level 4
  - Slow peripherals: UART, timers, SPI, I2C, GPIO
  - F ~ Fkernel / 1000

OMAP Bus Hierarchy



OMAP Global Memory Map (1)

AARHUS SCHOOL OF ENGINEERING

Table 1-1. Global Memory Space Mapping

QUARTER	Device Name	Start Address (HEX)	End Address (HEX)	Size	Description
Q0 (1GB)	**Boot space[1] GPMC**			**1MB 1GB or 1GB-1MB**	
	GPMC	0x0000 0000	0x3FFF FFFF	1GB	8/16 Ex[2]/R/W
Q1 (1GB)	**On-Chip Memory**			**128MB**	**ROM/SRAM address space**
	Boot ROM internal[1]	0x4000 0000	0x4001 3FFF	80KB	32-bit Ex[2]/R – Secure
		0x4001 4000	0x4001 BFFF	32KB	32-bit Ex[2]/R – Public
	Reserved	0x4001 C000	0x400F FFFF	912KB	Reserved
	Reserved	0x4010 0000	0x401F FFFF	1MB	Reserved
	SRAM internal	0x4020 0000	0x4020 FFFF	64KB	32-bit Ex[2]/R/W – Secure/public[3]
	Reserved	0x4021 0000	0x4024 FFFF	256KB	Reserved
	Reserved	0x4025 0000	0x47FF FFFF	128,704KB	Reserved
	L4 interconnects			**128MB**	**All system peripherals**
	L4-Core	0x4800 0000	0x48FF FFFF	16MB	See Table 1-3.
	(L4-Wakeup)[4]	(0x4830 0000)	(0x4833 FFFF)	(256KB)	(See Table 1-4.)

- Spuf98c chapter 2.1.2

OMAP Global Memory Map (2)

AARHUS SCHOOL OF ENGINEERING

	L3 Interconnect			**128MB**	**Control Registers**
	L3 Control Registers	0x6800 0000	0x68FF FFFF	16MB	See Table 1-2.
	Reserved	0x6900 0000	0x6BFF FFFF	48MB	Reserved
	SMS registers	0x6C00 0000	0x6CFF FFFF	16MB	Configuration registers SMS address space 2
	SDRC registers	0x6D00 0000	0x6DFF FFFF	16MB	Configuration registers SMS address space 3
	GPMC registers	0x6E00 0000	0x6EFF FFFF	16MB	Configuration registers GPMC address space 1
	Reserved	0x6F00 0000	0x6FFF FFFF	16MB	Reserved
Q2 (1GB)	**SDRC / SMS**			**1GB**	**SDRAM main address space (SMS)**
	CS0 – SDRAM (5)	0x8000 0000	0x9FFF FFFF	512MB	SDRC / SMS
	CS1 – SDRAM (5)	0xA000 0000	0xBFFF FFFF	512MB	SDRC / SMS

- Spuf98c chapter 2.1.2

OMAP L3 Memory Map

AARHUS SCHOOL OF ENGINEERING

Table 1-2. L3 Control Register Mapping

Device Name	Start Address (HEX)	End Address (HEX)	Size	Description
L3 RT	0x6800 0000	0x6800 03FF	1KB	L3 configuration registers
L3 SI	0x6800 0400	0x6800 07FF	1KB	Sideband signals configuration
Reserved	0x6800 0800	0x6800 13FF	3KB	Reserved
MPU SS IA	0x6800 1400	0x6800 17FF	1KB	MPU subsystem instruction port agent configuration
IVA2.2 SS IA	0x6800 1800	0x6800 1BFF	1KB	IVA2.2 subsystem initiator port agent configuration
SGX SS IA	0x6800 1C00	0x6800 1FFF	1KB	SGX subsystem initiator port agent configuration
SMS TA	0x6800 2000	0x6800 23FF	1KB	SMS target port agent configuration
GPMC TA	0x6800 2400	0x6800 27FF	1KB	GPMC target port agent configuration
OCM RAM TA	0x6800 2800	0x6800 2BFF	1KB	OCM RAM target port agent configuration

- Spuf98c chapter 2.1.3.1

OMAP L4 Memory Map

AARHUS SCHOOL OF ENGINEERING

Table 1-3. L4-Core Memory Space Mapping (1)

Device Name	Start Address (HEX)	End Address (HEX)	Size	Description
L4-Core	0x4800 0000	0x48FF FFFF	16MB	
Reserved	0x4800 0000	0x4800 1FFF	8KB	Reserved
System control module	0x4800 2000	0x4800 2FFF	4KB	Module
	0x4800 3000	0x4800 3FFF	4KB	L4 interconnect
Clock manager • DPLL • Clock manager	0x4800 4000	0x4800 5FFF	8KB	Module region A
	0x4800 6000	0x4800 67FF	2KB	Module region B
	0x4800 6800	0x4800 6FFF	2KB	Reserved
	0x4800 7000	0x4800 7FFF	4KB	L4 interconnect
Reserved	0x4800 8000	0x4802 3FFF	112KB	Reserved
Reserved	0x4802 4000	0x4802 4FFF	4KB	Reserved
	0x4802 5000	0x4802 5FFF	4KB	Reserved
Reserved	0x4802 6000	0x4803 FFFF	104KB	Reserved
I2C3	0x4806 0000	0x4806 0FFF	4KB	Module
	0x4806 1000	0x4806 1FFF	4KB	L4 interconnect
USBTLL module	0x4806 2000	0x4806 2FFF	4KB	Module
	0x4806 3000	0x4806 3FFF	4KB	L4 interconnect
HS USB HOST	0x4806 4000	0x4806 4FFF	4KB	Module
	0x4806 5000	0x4806 5FFF	4KB	L4 interconnect
Reserved	0x4806 6000	0x4806 9FFF	16KB	Reserved
UART1	0x4806 A000	0x4806 AFFF	4KB	Module
	0x4806 B000	0x4806 BFFF	4KB	L4 interconnect
UART2	0x4806 C000	0x4806 CFFF	4KB	Module

- Spuf98c chapter 2.1.3.2.1

OMAP GPMC Memory Map

AARHUS SCHOOL OF ENGINEERING

Table 1-28. GPMC Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address	Section
GPMC_SYSCONFIG	RW	32	0x0000 0010	0x6E00 0010	Section 1.1.7.2.1
GPMC_SYSSTATUS	R	32	0x0000 0014	0x6E00 0014	Section 1.1.7.2.2
GPMC_IRQSTATUS	RW	32	0x0000 0018	0x6E00 0018	Section 1.1.7.2.3
GPMC_IRQENABLE	RW	32	0x0000 001C	0x6E00 001C	Section 1.1.7.2.4
GPMC_TIMEOUT_CONTROL	RW	32	0x0000 0040	0x6E00 0040	Section 1.1.7.2.5
GPMC_ERR_ADDRESS	RW	32	0x0000 0044	0x6E00 0044	Section 1.1.7.2.6
GPMC_ERR_TYPE	RW	32	0x0000 0048	0x6E00 0048	Section 1.1.7.2.7
GPMC_CONFIG	RW	32	0x0000 0050	0x6E00 0050	Section 1.1.7.2.8
GPMC_STATUS	RW	32	0x0000 0054	0x6E00 0054	Section 1.1.7.2.9
GPMC_CONFIG1_i (1)	RW	32	0x0000 0060 + (0x0000 0030 * I)	0x6E00 0060 + (0x0000 0030 * I)	Section 1.1.7.2.10
GPMC_CONFIG2_i (1)	RW	32	0x0000 0064 + (0x0000 0030 * I)	0x6E00 0064 + (0x0000 0030 * I)	Section 1.1.7.2.11
GPMC_CONFIG3_i (1)	RW	32	0x0000 0068 + (0x0000 0030 * I)	0x6E00 0068 + (0x0000 0030 * I)	Section 1.1.7.2.12

- Spuf98c chapter 11.1.1.7.1