

iCE65 as CellularRAM Controller

Overview

CellularRAM memory is the best option for low to mid-range feature-rich mobile phones due to its lowest power and low cost per bit features. This design example illustrates the implementation of a CellularRAM controller using iCE65 FPGAs. As a CellularRAM controller, the iCE65 FPGA generates all the required interfacing signals and implements the functions that are required for such a system.

Description

CellularRAM is an enhanced PSRAM, which combines SRAM's minimal power consumption with DRAM's high bandwidth. It provides asynchronous and burst mode read/write with performance enhancing features such as variable latency. CellularRAM memory has a short bit line structure, enabling fast random access times of the order of 70 to 85ns and performance with burst speeds of up to 104MHz.

Power-Up Initialization

CellularRAM products include an on-chip voltage sensor used to launch the power-up initialization process, which configures BCR(Bus Control Register) and the RCR(Refresh Configure Register) with their default settings. Self-initialization process requires 150 μ s to complete, during that period Chip Enable(CE#) should remain HIGH.

Deep Power-Down Mode (DPD)

DPD mode disables all refresh-related activity there by reducing lowest power consumption (current<3 μ A). This mode is used if the system does not require the storage provided by the CellularRAM device. Any stored data will become corrupted when DPD is enabled. DPD can be enabled by writing to the RCR using CRE or the software access sequence; DPD starts when CE# goes HIGH.

Bus Operating Modes

CellularRAM bus interface supports both asynchronous and burst mode transfers. CellularRAM 1.5 products power up in the asynchronous operating mode.

Asynchronous Mode

This mode uses the industry-standard SRAM control bus (CE#, OE#, WE#, LB#/UB#). READ operations are initiated by bringing CE#, OE#, and LB#/UB# LOW while keeping WE# HIGH. Valid data will be driven out of the I/Os after the specified access time has elapsed.

WRITE operations occur when CE#, WE#, and LB#/UB# are driven LOW. During asynchronous WRITE operations, the OE# level is a "Don't Care," and WE# will override OE#. The data to be written is latched on the rising edge of CE#, WE#, or LB#/UB# (whichever occurs first).

During asynchronous operation, the CLK input must be held static LOW. WE# LOW time must be limited to tCEM(~70-85ns). Figure 1 and Figure 2 illustrates the Asynchronous mode Read and Write operations.

Burst Mode Operation

Burst mode operations enable high-speed synchronous READ and WRITE operations. After CE# goes LOW, the address to access is latched on the rising edge of the next clock that ADV# is LOW. During this first clock rising edge, WE# indicates whether the operation is going to be a READ (WE# = HIGH) or WRITE (WE# = LOW). Figure 3 and 4 illustrates Burst mode Read and Write operations.

The size of a burst can be specified in the BCR either as a fixed length or continuous. Fixed-length bursts consist of 4, 8, 16, or 32 words. The latency count stored in the BCR defines the number of clock cycles that elapse before the initial data value is transferred between the processor and CellularRAM device. CellularRAM devices supports both fixed and variable latency mode. Fixed latency outputs the first data word after the worst-case access delay, including allowance for refresh collisions. The WAIT output asserts when a burst is initiated and de-asserts to indicate when data is to be transferred into (or out of) the memory.

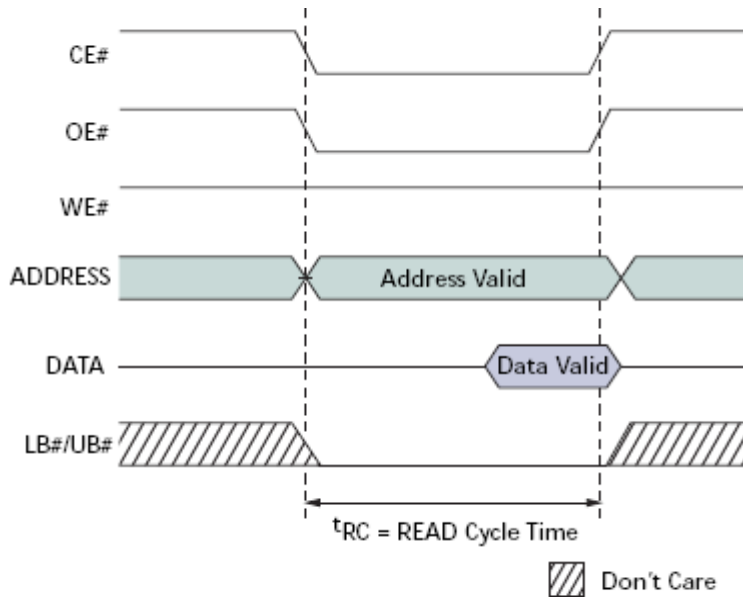


Fig 1: Asynchronous mode read cycle

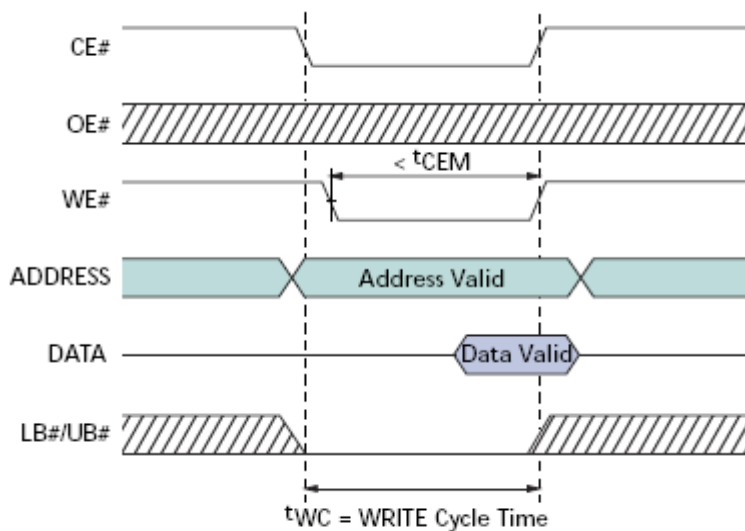


Fig 2: Asynchronous mode write cycle

Implementation

The CellularRAM Interface implementation is based on CellularRAM specification version 1.5. Figure 5 shows the overall interface of CellularRAM controller to both host processor and CellularRAM device. The controller FSM block monitors host side transaction requests to the CellularRAM through CPU interface logic. Depending on the host processor request, the appropriate control signals are sent to the CellularRAM along with the address and data. The controller FSM is the heart of the CellularRAM controller which makes sure that correct transactions take place in sequence and that the CellularRAM devices timings are not violated in the process. Figure 6 shows a simplified diagram of the state machine.

1. Power-on brings the CellularRAM controller to power-up-initialization state and by default CellularRAM goes to Asynchronous mode of operation. CS must be held High during power-on.

2. Host uses read, write, register_addr and data_bus pins to configure CellularRAM controller to desired mode and configuration.

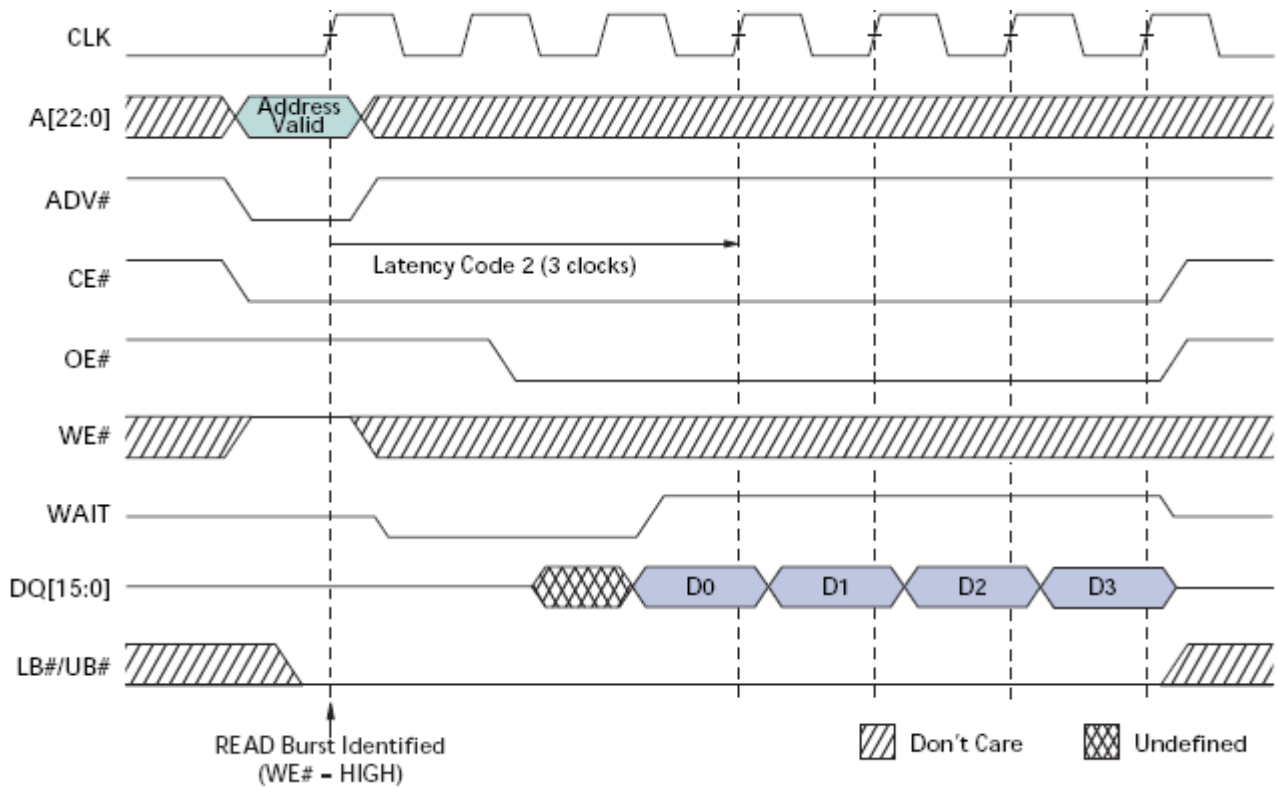


Fig 3: Burst mode read cycle(4 word)

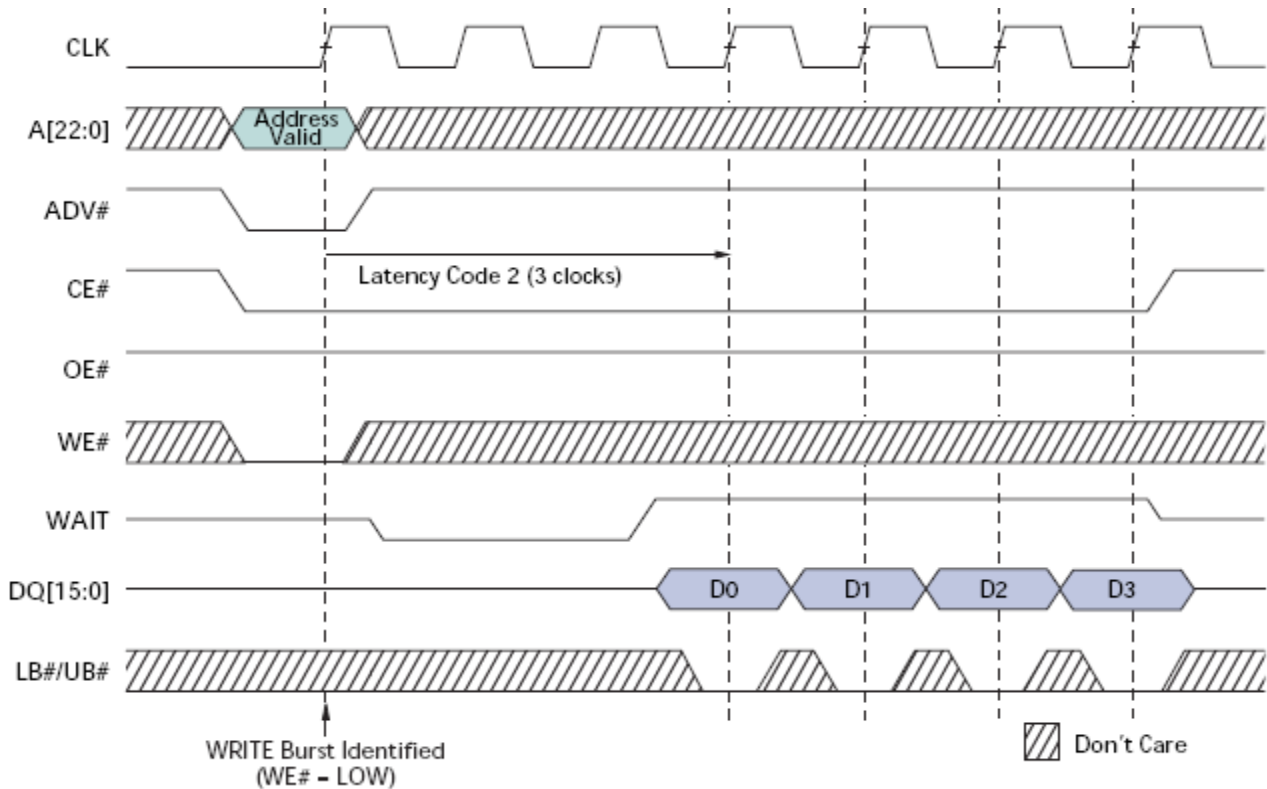


Fig 4: Burst mode write cycle(4 word)

3. {read, write, register_addr} signals configures read/write operations from/to host CPU as follows:

- when "1-1100", writes data from data_bus to fifo module
- when "1-111", reads fifo contents to data_bus
- when "01101", writes data bus contents to configuration register
- when "10110", reads status register contents to data_bus
- when "1-1000", CPU data_bus is connected to FIFO data_in but write enable is low. Used as stable state during write operations
- when "1-0111", CPU data_bus is connected to FIFO data_out but read enable is low. Used as stable state during read operations

4. FIFO(first in first out) module controls all the read and write data transactions that takes place between CPU and CRAM . There are two separate FIFO modules. One for writing and other for reading operations with respect to CPU. CPU addresses the FIFO either for reading or writing the data by giving the appropriate read, write, register_addr signals. Read and write enables are active high. fifo_empty and fifo_full status information from status registers informs the CPU whether the write FIFO or read FIFO is empty or full respectively. Write or read enables should be made low when these signals are high

5. Configuration register bits program the C-RAM controller mode, burst length and drive strength as follows:

- ram mode set to asynchronous read/write when configuration register(0) is '0', otherwise set to burst mode.
- config_reg(4 downto 2) decides the burst length as per cellular RAM specifications.
- config_reg(6 downto 5) decides the drive strength as per cellular RAM specifications
- config_reg(9 downto 7) decides the burst latency as per cellular RAM specifications
- config_reg(15) decides burst stop mode. A high on this bit stops burst read/write operation and a low for resuming it
- config_reg(14) decides power down mode. A high on this bit enables the DPD mode and a low on this bit can be used to recover from DPD state

6. Combination of {pwrdrn, ram_mode, read, write, burst_stop} signals from host CPU issues following commands to CellularRAM controller:

- "1---" - Power down mode
- "0-110" - Stand-by mode(NOP)
- "00000" - Asynchronous mode initialization
- "00010" - Asynchronous mode write
- "00100" - Asynchronous mode read
- "01000" - Burst mode initialization
- "01010" - Burst mode write
- "01100" - Burst mode read
- "01011/01101" - Burst stop mode

6. Interrupt from CRAM controller - CRAM controller generates a High on cram_intr pin for following conditions - write fifo full, read fifo full, DPD, Cellular RAM read/write completion.

7. Status register contents provide following C-RAM controller status : {NONE, NONE, write fifo full, write fifo half full, write fifo empty, read fifo full, read fifo half full, read fifo empty, NONE, NONE, cram busy, DPD acknowledge, write acknowledge, read acknowledge, configuration acknowledge, power up status}.

Table 1 lists the pin description of CellularRAM controller.

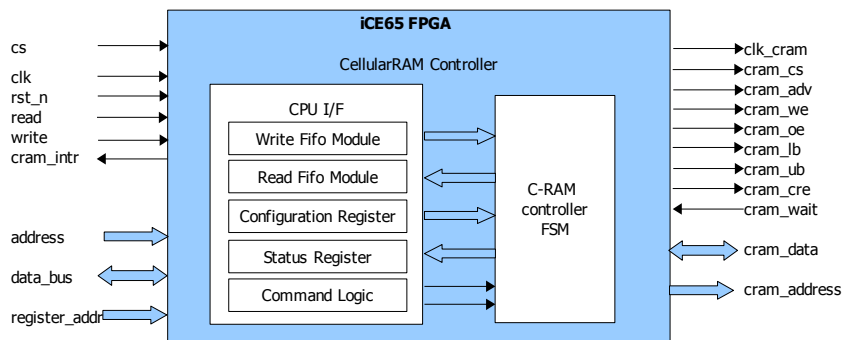


Fig 5: Block Diagram of CellularRAM controller

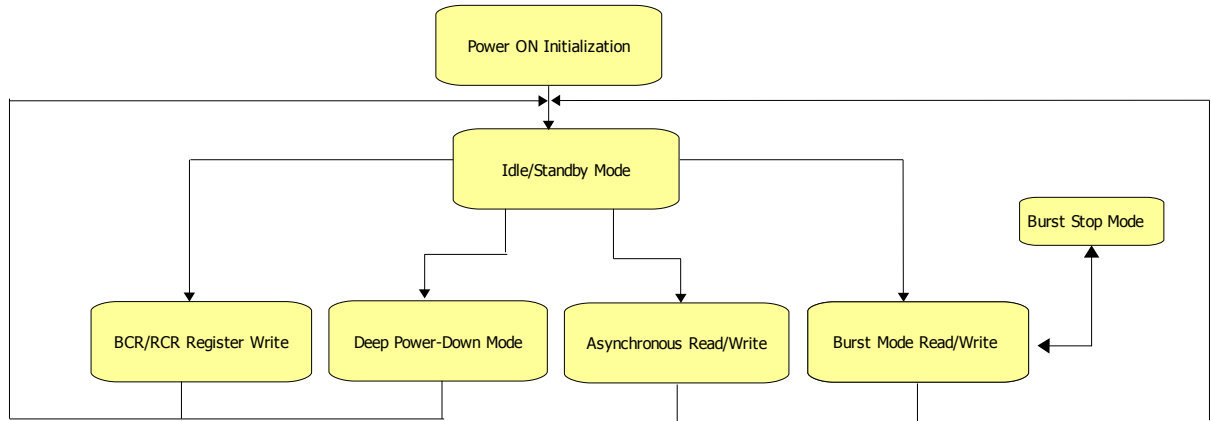


Fig 6: Simplified State Diagram

| | Direction | Description |
|--------------------|-----------|--|
| cs | Input | C-RAM controller select. Active High |
| clk | Input | System Clock |
| rst_n | Input | System Reset, Active Low |
| read | Input | Memory Read, Active High |
| write | Input | Memory Write, Active High |
| address[22:0] | Input | Host Address Bus |
| cram_intr | Output | Asserts high when generates an interrupt to host |
| data_bus[15:0] | Inout | Host Data Bus |
| register_addr[2:0] | Input | Register Select |
| clk_cram | Output | C-RAM clock |
| cram_cs | Output | C-RAM chip select |
| cram_adv | Output | C-RAM Address valid |
| cram_we | Output | C-RAM write enable |
| cram_oe | Output | C-RAM Output Enable |
| cram_lb | Output | C-RAM Lower byte enable |
| cram_ub | Output | C-RAM Upper byte enable |
| cram_cre | Output | C-RAM Configure Register Enable |
| cram_wait | Input | C-RAM Wait state |
| cram_data[15:0] | Inout | C-RAM data bus |
| cram_address[22:0] | Output | C-RAM address bus |

Table 1: Pin Description

Table 2 summarizes the post P&R resource utilization summary of this Design Example when implemented using iCE65 FPGA.

| Device | Logic Cells | IO Cells |
|-----------------|-------------|----------|
| iCE65L04-UCB284 | 753 | 96 |

Table 2: Resource Utilization

Conclusion

This design example demonstrates the implementation of a CellularRAM controller using iCE FPGAs. The low power, high density and speed properties of CellularRAMs are well complemented by iCE FPGA's very low power and high speed capabilities. This makes iCE FPGAs an obvious choice for CellularRAM controller in all low power and portable applications like feature-rich cellphones and smartphones.

Disclaimer

SiliconBlue is providing this document, design example, or information "as is." SiliconBlue does not ensure that this implementation or information is void of any claims of infringement. As a reader / implementer, you are responsible for obtaining any rights you may require for your implementation. SiliconBlue also does not warranty the fitness of this design example to be readily implemented in any specific context. It is your sole responsibility to validate this design example and its correctness while implementing it for your requirement.