

Wasca memory maps v. 0.4

This is a short document describing max10-based wasca cartridge memory and registers map from both SH-2 and in-FPGA Nios II soft cpu perspective. It also covers some aspects about developer Zynq-based board (this one is not named yet).

Part 1. SH-2 memory map

All the absolute addresses in this part of the document are in SH-2 cache-through address space unless otherwise specified.

Table 1-1 describes differences between SH-2's chipselect spaces and A-bus (SMPC's chipselects). Further chipselect references will be using A-Bus ranges.

Address range	Size	SH-2 chipselects	A-Bus Chipselects	wasca usage
0x22000000 - 0x23FFFFFFF	32 MB	CS0	CS0	Yes
0x24000000 - 0x24FFFFFFF	16 MB	CS1	CS1	Yes
0x25000000 - 0x257FFFFFFF	8 MB		Dummy	No
0x25800000 - 0x258FFFFFFF	1 MB		CS2	Not yet

Table 1-1. SH-2 and A-bus chipselect ranges.

Table 1-2 shows a detailed map of CS0 and CS1 and its usage by some official and unofficial carts.

Address range	Size	CS	Power memory	RAM cart	ROM cart	Bootable cart
0x22000000 - 0x221FFFFFFF	2 M	CS0			Data	Code and data
0x22200000 - 0x223FFFFFFF	2 M					
0x22400000 - 0x225FFFFFFF	2 M			Lower data		
0x22600000 - 0x227FFFFFFF	2 M			Upper data		
0x22800000 - 0x23FFFFFFF	24 M					
0x24000000 - 0x24FEFFFF	15.9999 M	CS1	Data			
0x24FFFFFFC	4		ID	ID	0xFFFF	0xFFFF

Table 1-2. Detailed CS0 and CS1 memory maps for various carts.

Table 1-3 shows the same map of CS0 and CS1 and its usage by wasca. Empty areas in tables 1-2 and 1-3 are unmapped, and should return unpredictable value (mostly 0xFFFF) on every read access. Write accesses in these areas are ignored.

Address range	Size	CS	wasca mode						
			INIT	Power memory	RAM	<32M ROM	32M ROM		
x22000000 - 0x221FFFFFFF	2 M	CS0	Bootcode			ROM	ROM		
0x22000000 - 0x223FFFFFFF	2 M								
0x22400000 - 0x225FFFFFFF	2 M				RAM				
0x22600000 - 0x227FFFFFFF	2 M								
0x22800000 - 0x23FEFFFF	23.94 M								
0x23FF0000 - 0x23FFDFFF	56K		Reserved						
0x23FFE000 - 0x23FFEFFF	4K		Filesystem interface						
0x23FFF000 - 0x23FFFFFFF	4K		wasca system area						
0x24000000 - 0x24FE0FFF	15.94 M	CS1		Power memory					
0x24FF0000 - 0x24FFDFFF	56K						Reserved		
0x24FFE000 - 0x24FFEFFF	4K						File system interface		
0x24FFF000 - 0x24FFFFFFB	4092						wasca system area		
0x24FFFFFFC (ID)	4			0xFF	0x2x	0x5x	0xFF	0xFF	

Table 1-3. Detailed CS0 and CS1 memory maps.

Filesystem interface is introduced to allow SH2 browse SD card or other media content, read, write or select files there. Selecting file is required for custom ROM operation mode. Filesystems registers are 16-bit-wide and should be accessed with 16-bit accesses and 16-bit address align. **Other access sizes and/or unaligned addresses will lead to unpredictable behavior.**

Filesystem is locked after reset, and should be unlocked by writing 0xFADE into LOCK register. While locked, filesystem interface behaves as an unmapped area, LOCK register is write-only.

Offset	Size	R/W	Name	Description
0x000	2048	R/W	DATA BUFFER	Buffer for storing data for write and read operations.
0x800	2032	R/W	COMMAND BUFFER	Buffer for storing commands and replies. SH-2 software should write a valid command here before writing to bit 0 of the COMMAND register. Commands should be UTF8-encoded. When bit 0 in STATUS register reports a command completion, SH-2 should expect result here. Error messages requested by bit 1 in the command register appears here as well. SH-2 should not read this buffer while busy bit in STATUS register is active.
0xFF0	2	R/W	LOCK	Write 0xFADE to this register to unlock filesystem functionality. Filesystem control is locked after every reset

				and should be unlocked by SH2 software. Writing any other value into this register enables lock again.
0xFF2	1	W	COMMAND	SH-2 software should write '1' to bit 0 to start a command stored in the command buffer beforehand. To get last error message, SH-2 software should write '1' to bit 1. Other bits are reserved, and should be written as zeros.
0xFF4	2	R	STATUS	Bit 0 : busy bit, 1 when command is processed, 0 when done. Bit 1 : error bit, flags if any errors were detected during the last command. Other bits are reserved and should be ignored.

Table 1-4. Filesystem interface.

Filesystem interface support a reduced set of unix-shell-like commands. **Please note that the command syntax and replies structure are not unix-shell compatible, do not use a man pages or other general documentation as a reference.** Supported commands are enclosed in table 1-5. <0> are zero-terminators.

File and folder names could be either relative (e.g. “sound/sega.wav”), or absolute (e.g. “/sdcard/media/sound/sega.wav”). If no path provided for a filename, current folder is used.

Command	Syntax	Description
cd	cd <folder><0>	Enters a specified folder. If “..” is provided as a folder name, leaves current folder.
close	close <0>	Closes currently opened file.
cp	cp <filename 1> <filename 2><0>	Copies file1 to file2. If file2 exists, returns error and does nothing.
ls	ls<0>	Lists current folder. Filename are is stored in the COMMAND BUFFER, separated by zero-symbols. List is ended by double zero. Due to limited command buffer size, command could return an incomplete list, to get further files in the folder SH-2 software should issue ls command again. When the command returns a zero list (started by a two zero-symbols), there are no more files to list. Issuing any other command resets this one to the start of the folder.
mv	mv <filename 1> <filename 2><0>	Moves file1 to file2. If file2 exists, returns error and does nothing.
open	open <filename> [<access>]<0>	Opens a file for reading or writing. Valid access fields are “r” (read-only, default), “w” (read-write), “n” (read-write, create file, truncate file if exists). If the file does not exist and the access field is not “n”, error is returned. Only 1 file can be opened at a time, issuing open when a file is already open closes currently open file.
pwd	pwd<0>	Returns current path
read	read <number><0>	Reads a number of bytes into DATA BUFFER. Returns error if number is more than 2048 or no file is currently opened.
rm	rm <filename><0>	Deletes a file from the filesystem. If file does not exist, returns error.
seek	seek <number><0>	Sets a data pointer in a file to the position provided by

		number. Poiter is automatically set to 0 when opening file, and auto-increases when reading or writing, so seeking could be avoided when accessing file from the start to the end.
stat	stat <filename><0>	Returns a detailed file characteristics, like size, permissions, user, group etc. in the text format similar to 'ls -l' in linux.
write	write <number><0>	Writes a number of bytes from DATA BUFFER. Returns error if number is more than 2048 or no file is currently opened.

Table 1-5. Filesystem commands.

Table 1-6 describes wasca system area. System wasca registers are 16-bit-wide and should be accessed with 16-bit accesses and 16-bit address align. **Other access sizes and/or unaligned addresses will lead to unpredictable behavior.** Address range for system area is remapped in 32M ROM mode.

Offset	Size	R/W	Name	Description
0x000	4080		Reserved	
0xFF0	2	R	PCNTR	Wasca Prepare counter. Reset value is 0, after writing a valid value to MODE, this counter shows value (in percents) of the prepare progress. When this register has value of 0x0064, wasca is ready to emulate specified cart. Higher values should not appear normally.
0xFF2	2	R	STATUS	Wasca Status register. Error and flag bits appears here. Bit 0 – SD card error when 1 Bit 1 – SDRAM test failed when 1 Bit 2 – USB transfer error when 1 Other bits are reserved
0xFF4	2	R/W	MODE	wasca Mode Register Divided into 4 octets, non-zero octet selects cart type, octet value selects card subtype. See table 5 for additional description. Setting values not provided in table 5 results in unpredictable behavior.
0xFF6	2	R	HWVER	wasca hardware version, major and minor 0x050C = v5.12
0xFF8	2	R	SWVER	wasca embedded software version, major and minor (same format as HWVER) This register is zero at reset, and is written by cart's firmware right after boot. If by the time SH-2 is running the bootcode this value is still zero, it means cart failure. SH-2 cannot alter this register.
0xFFA	6	R	SIGNATURE	For max10 “wasca “ in ASCII (0x7761 0x7363 0x6120), for snickerdoodle “doodle” in ASCII (0x646F 0x6F64 0x6C5C). Signature unavailable in remapped (32M ROM) mode.

Table 1-5. Wasca system area registers.

Okay, and now the promised wasca modes table:

MODE value	Description
0x0000	INIT
0x0001	Power memory, 0.5 MB
0x0002	Power memory, 1 MB
0x0003	Power memory, 2 MB
0x0004	Power memory, 4 MB
0x0020	RAM expansion, 1 MB
0x0040	RAM expansion, 4 MB
0x0100	ROM, KoF95.bin
0x0200	ROM, Ultraman.bin
0x0F00	ROM, custom. Actual ROM file is selected via filesystem interface by SH-2. If the file size is 32M, wasca registers are remapped according to the table 1-3.
Other values	Unpredictable

Table 1-5. Valid wasca MODE register values.

Part 2. Nios II memory map

Nios II address range is FPGA-dependent (will be different for different FPGA parts) and is still work in progress, so it might change occasionally. Please check the last wasca FPGA project from repository for the valid memory map. Table 2-1 shows a memory map for Nios II soft cpu.

Address range	Size	Description
0x00000000 - 0x0002AFFF	172 KB	On-chip flash. It stores both Nios II software and SH-2 bootcode (compiled into Nios II software as binary).
0x00040000 — 0x00040001F	32 B	UART registers
0x00041000 — 0x000417FF	2 KB	Debug RAM
0x00042000 — 0x0004200F	16 B	PLL setup
0x00043000 — 0x000433FF	1 KB	SD card interface
0x00044000 — 0x0004400F	16 B	Reserved
TBD	TBD	Filesystem interface
0x00045000 — 0x000451FF	512 B	wasca system registers
0x00080000 — 0x000827FF	10K	RAM

Table 2-1. Nios II memory map.

The scope of this document does not cover Nios II peripheral description, but it is well covered in Altera docs. The custom cores in Nios II memory map are filesystem interface and wasca system registers interface. Filesystem interface is TBD, wasca system area is shown in table 2-2.

Offset	Size	R/W	Name	Description
0x000	4080		Reserved	
0xFF0	2	R/W	PCNTR	Wasca Prepare counter. After sensing a non-zero value in MODE register Nios should start the prepare process, and write the progress periodically into this value (in percents).
0xFF2	2	R/W	STATUS	Wasca Status register. Nios should write status here. See corresponding register in SH-2 memory map.
0xFF4	2	R	MODE	wasca Mode Register Nios should scan this register while in INIT mode, and switch to corresponding mode when it changes. See corresponding register

				in SH-2 memory map.
0xFF6	2	R	HWVER	wasca hardware version, major and minor 0x050C = v5.12
0xFF8	2	R/W	SWVER	This register is zero at reset, and is written by Nios soft CPU right after boot. See corresponding register in SH-2 memory map.
0xFFA	6		Reserved	

Table 2-2. wasca system area from Nios II perspective.

Part 3. Zynq memory map

Zynq memory map is generated automatically by Vivado. After this Vivado generates macros for every IP core, and the actual address is tied to that macros. Therefore, in the scope of this document, only the relative addresses (within IP core) will be described.

CORE 1: A-Bus bridge.

Accessible by AXI4Lite bus for status and configuration. Uses the same address map as Nios II (see table 2-2).

CORE 2: Filesystem interface.

Accessible by AXI4Lite bus for data transfer. Uses the same address map as Nios II (TBD).

CORE 3: Snoop core.

Accessible by AXI4Lite bus for status and configuration. Snoop core catches transactions over A-Bus and records them as a list into DDR.

Offset	Size	R/W	Name	Description
0x000	4	R/W	SNOOP_ADDRESS	Selects address in an AXI space range for snooping A-BUS data transactions.
0x004	4	R/W	SNOOP_FILTER	Filter for snooping only desired transactions. Set the corresponding bit to 1 to enable. Bit 0 – CS0, Bit 1 – CS1, Bit 2 – CS2, Bit 3 – reads, Bit 4 – writes.
0x008	4	R/W	SNOOP_FORMAT	Snooping format as follows: 0x00 – 16 bit (1bit r/w, 15 bit lower address) 0x01 – 32 bit (1 bit r/w, 2 bit cs, 26 bit address, 3 bit hold length) 0x02 – 64 bit (1 bit r/w, 2 bit cs, 26 bit address, 16 bit data, 3 bit hold length, 1 bit hold ignore, 15 bit timestamp)
0x00C	4	R/W	SNOOP_CONTROL	Bit 0 enables and disables snoop. Bit 1 resets snoop to start address and clears itself.
0x010	4	R	SNOOP_COUNTER	Holds value for current number of transactions snooped. Resets to 0 after setting bit 1 in SNOOP_CONTROL

CORE 4: Display core.

Accessible by AXI4Lite bus for data transfer. Display sports its own internal video buffer, so the data in the core's memory is not displayed upon writing, but should be uploaded from core into display first. To do this one should either upload data manually by writing 1 to DISPLAY_CONTROL register, or enable auto-upload.

Offset	Size	R/W	Name	Description
0x00000	0x12C00	R/W	FRAMEBUFFER	Framebuffer for holding pixel data.
0x1F000	1	R/W	DISPLAY_CONTROL	Bit 0 – uploads framebuffer upon writing 1 Bit 1 – enables automatic upload

0x1F004	1	R	UPLOAD_STATUS	Bit 0 – upload is active.
0x1F008	4	R/W	UPLOAD_INTERVAL	Automatic upload interval in microseconds.
0x1F00C	4	R	UPLOAD_COUNTER	This counter cyclically counts from 0 to UPLOAD_INTERVAL, then resets to 0 and starts over. Upload occurs when UPLOAD_COUNTER is 0. Upload takes some time, software should check UPLOAD_STATUS register before writing to buffer.
0x1F010	1	R/W	BACKLIGHT	Backlight intensity. 0x00 – minimal, 0xFF -maximal.

Part 4. Revision history

v0.1	Preliminary version, system regs and SD card interface, only SH-2 perspective
v0.2	Nios perspective added, some additional comments
v0.3	SD card interface removed, filesystem interface added, Zynq perspective added
v0.4	Filesystem interface changed drastically