RW-WLAN HE TOP CPU Integration Guidelines

Integration Guide
RW-WLAN-HE-TOP-CPU-HW-IG
Version 0.01
2018-12-31

RivieraWaves
http://www.rivierawaves.com/

Revision History

Version	Date	Revision Description	Author
0.01	2018-12-18	Updated with MAC and Platform information	Jerome Vanthournout

Table of Contents

List of Figures

List of Tables

1 Overview

This document gives guidelines for integration of the RW-WLAN HE TOP CPU IP. Note that more details information on the clock generation and memories are provided in [5].

It covers:

- Modem configuration guidelines for the 1x1 use case
- Platform configuration guidelines for the 1x1 use case
- Interconnect guidelines:
  - I/O
  - Clocks
  - Resets
- Implementation guidelines:
  - Synthesis flow

2 Integration example

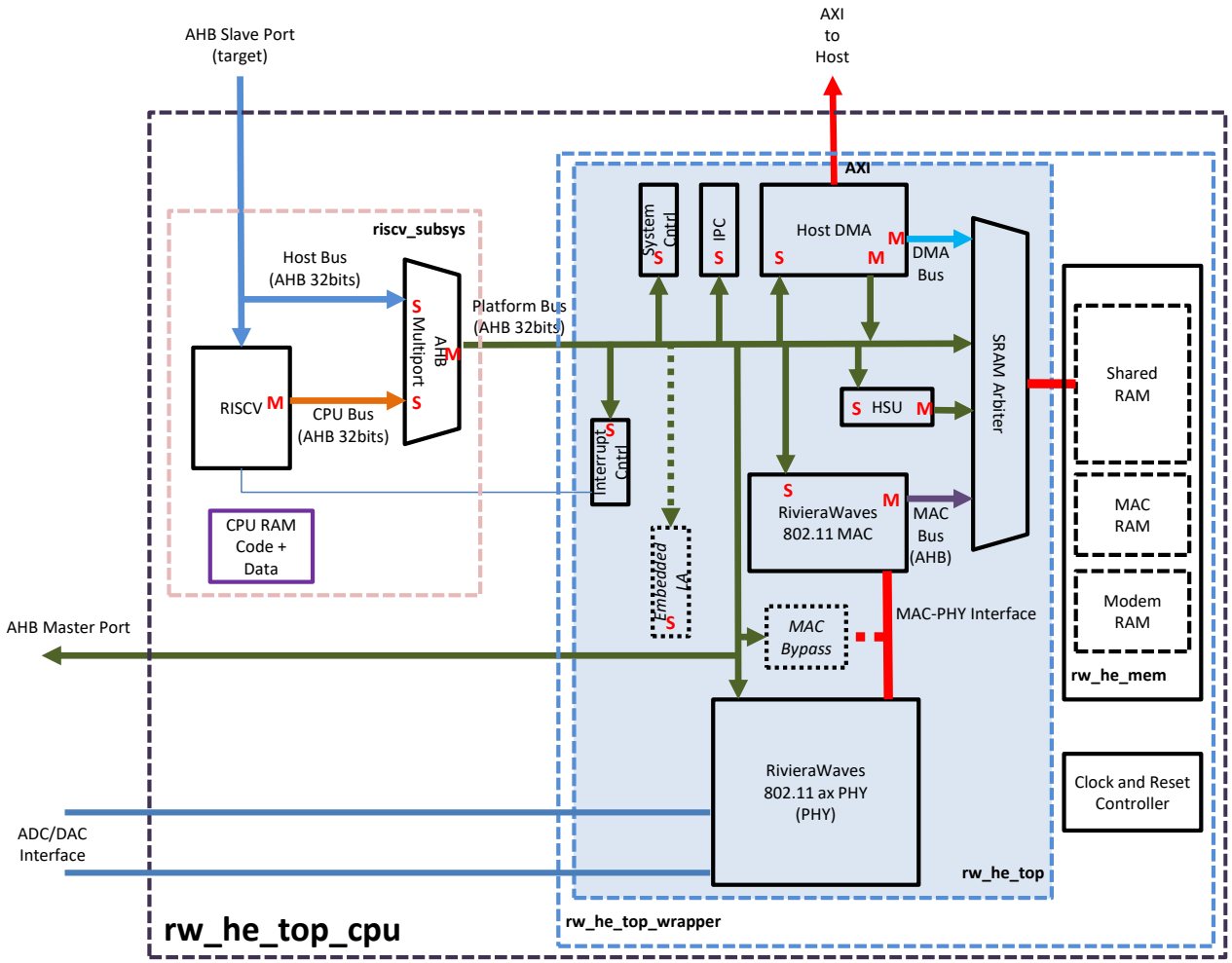

Figure 2-1: rw_he_top_cpu

The RW-WLAN HE TOP CPU IP is identified by the “rw_he_top_cpu” modules, and provides an integrated solution for the RW-WLAN modem and platform IPs, as illustrated in Figure 2-1. The Modem IP consists in the “hdm_core” and “RIUCore” modules. The Platform and MAC IP consists in an APH platform and an MAC HW Accelerator named rwWlanNxMACHW. In order to ease the IP integration, it is provided:

- An example of Clock/Reset controller “rw_he_crm” which indicates how the clocks and the resets have to be generated for correct operations. The Clock/Reset controller is found under <…>/IPs/HW/TOP11AX/rw_he_crm
- An example of memory models which indicates how the memory models must behave with the Modem IP modules. The models are found under <…>/IPs/HW/TOP11ax/HWCOMMON/memory_models and the integration example in <…>/IPs/HW/TOP11AX/rw_he_memories
- An example of integration of the RIUCore with a Radio Controller, found under <…>/SB/RIU_KARST/RIUKarstTop/verilog/rtl/RIUKarst.v
- An example of integration within a SOC which indicates how IP, clock/reset controller and memories model are interconnected. The example is found under <…>/IPs/HW/TOP11AX/rw_he_top/Verilog/rtl/rw_he_top_wrapper.v
- An example of integration within a RISC V CPU which indicates how IP, clock/reset controller and memories model are interconnected. The example is found under <…>/IPs/HW/TOP11AX/rw_he_top_cpu/verilog/rtl/rw_he_top_cpu.v

Besides,

- The “rw_he_top” module is found under <…>/IPs/HW/TOP11AX/rw_he_top
- The “hdm_core” module is found in the directory <…>/IPs/HW/TOP11ax/PHYSUBSYS/HDMCORE/hdm_core.
- The “RIUCore” module is found in <…>/IPs/HW/TOP11ax/PHYSUBSYS/MDMCOMMON/RIUCORE/RIUCore.
- The “rw_nx_platform” module is found in <…>/IPs/HW/TOP11ax/MACSUBSYS/rw_nx_platform.
- The “MAC HW” module is found in <…>/IPs/HW/TOP11ax/MACSUBSYS/MACCORE/rwWlanNxMACHW.

The WLAN HE TOP IP is a full synchronous design that applies the following rules:

- It does not contain any sequential cell other than D flops (no latch, no tie).
- All flops are rising edge.
- There is no logic on the asynchronous reset trees.
- There is no logic on the clock trees.
- It does not instantiate any black-box, synthesis will only infer the cells found in the target library.
- All data crossing between clock domains uses a unique single resynchronization module. The integrator can then choose to either synthesize this module (2 FFs), or replace it by a specific standard cell during synthesis.

These rules ensure that the design is independent of the library technology used, and facilitate the integration with our customer’s front-end, back-end and DFT flows.

However, the WLAN HE TOP IP implements some advanced features that require some simple circuitries beside the Modem IP to guarantee a successful integration:

- Multiple clock domains: Every clock domain displays a clock root input to the Modem IP. It is up to the integrator to derive and align these clocks from a 240 MHz source.
- Power reduction based on dynamic gated clocks: the modem asserts clock enable signaling. It is up to the integrator to gate these clocks with the appropriate timing relationship.
- Dynamic clock switching: the modem asserts clock switch signals. It is up to the integrator to multiplex these clocks with the appropriate timing relationship.
- Dynamic asynchronous reset: the modem dynamically asserts a reset request in order to re-initialize the DSP domain through its asynchronous reset tree. It is up to the integrator to handle this asynchronous reset signaling.

3 IP Configuration

The WLAN HE TOP IP RTL code is configurable through parameters in order to implement the different modem configurations from a single source code:

- Number of transmit streams: $N_{TX} = 1$
- Number of receive streams: $N_{RX} = 1$
- Bandwidth mode (support of 20 MHz only or 20/40 MHz)
- Support of LDPC
- Support of 11ac Beamforming

These parameters are defined in a user project define file, e.g. in WLAN_HE_REF_IP/HW/env/CONF. They are described in Table 3-1.

Parameter Name	Value	Description
RW_NX_CHBW20	Only 1 of the 2 shall be defined	When defined, only 20MHz operation is supported
RW_NX_CHBW4020		When defined, only 40 and 20MHz operation are supported
RW_TXRX_1X1		Antenna configuration: 1 Tx antennae, 1 Rx antennae
RW_RADAR_EN		When defined, radar detection is included
RW_NX_LDPC_ENC		When defined, LDPC FEC mode is supported in transmit
RW_NX_LDPC_DEC		When defined, LDPC FEC mode is supported in receive
RW_NX_256QAM_EN		When defined, 256QAM is supported in HE
RW_BFMEE_EN		When defined, beamformee support is included
RW_CONFIG_6VAP_40STA		Number of supported STA (6 VAPs and 40 STAs). This is the lowest configuration for STA.
RW_GCMP_EN		When defined. Enabled the GCMP support
RW_WAPI_EN		When defined. Enabled the WAPI support
RW_WLAN_COEX_EN		When defined. Enabled the Coexistence support
MAC_FREQ		Define the MAC Core frequency
MACPI_CLK_FREQ		Define the Platform frequency
WEP_2_BB_CLK_RATIO		Define the ratio between macCoreClk and wtClk. If MAC_FREQ is at least 60MHz, this ratio shall be set to 1

Table 3-1: RW-WLAN HE TOP IP configuration

Derived parameters, defined in <…>/IPs/HW/TOP11AX/PHYSUBSYS/HDMCORE/cfg/hdm_core_config.v, are used in the Modem's blocks RTL code to define their internal bus size and the numbers of identical sub-blocks that needs to be instantiated as described in [1].

Derived parameters, defined in <…>/IPs/HW/TOP11AX/MACSUBSYS/MACCORE/rwWlanNxMACHW/conf/define.v, are used in the MAC blocks RTL code to define their internal bus size and the numbers of identical sub-blocks that needs to be instantiated as described in [3].

4 Interconnect guidelines

The following guidelines apply with the following configuration: 1 RX, 1 TX and a channel bandwidth of 20/40MHz.

4.1 Clock connections

Clock pin	Description
ref0_root_clk	Reference clock for PHY (240/480 Mhz)
ref1_root_clk	Not used
plf_root_clk	Platform clock
mac_lp_root_clk	Low-Power clock

Table 4-1: clock signals

4.2 Asynchronous resets

Each main clock domain is associated to an asynchronous reset. Asynchronous reset can be asserted at any time but must be de-asserted synchronously.

The table below associates the different reset with clock domains:

Reset port	Applicable clocks	Associated HW reset request port
sys_rst_n	ref0_root_clk	Global Reset

Table 4-2: reset signals

4.3 Port map

Name	Type	Size	Description
AHB Slave interface			
target_hready	output	1	Processor hready
target_hready_in	input	1	Processor hready_in
target_haddr	input	32	Processor haddr
target_htrans	input	2	Processor htrans
target_hwrite	input	1	Processor hwrite
target_hmastlock	input	1	Processor hmastlock
target_hburst	input	3	Processor hburst
target_hprot	input	4	Processor hprot
target_hsize	input	3	Processor hsize
target_hrdata	output	32	Processor hrdata
target_hwdata	input	32	Processor hwdata
target_hresp	output	2	Processor hresp
host_irq	output	1	Interrupt from host
GPIO interface			
gpio_out	output	32	GPIO output
gpio_in	input	32	GPIO input
gpio_oen	output	32	GPIO output enable
AHB to Coexistence interface			

coexif_hready_in	output	1	Coexistence Interface hready_in
coexif_hsel	output	1	Coexistence Interface hsel
coexif_haddr	output	9	Coexistence Interface haddr
coexif_htrans	output	2	Coexistence Interface htrans
coexif_hwrite	output	1	Coexistence Interface hwrite
coexif_hrdata	input	32	Coexistence Interface hrdata
coexif_hwdata	output	32	Coexistence Interface hwdata
coexif_hresp	input	2	Coexistence Interface hresp
coexif_hready	input	1	Coexistence Interface hready
Bluetooth Coexistence Interface			
coex_bt_tx	input	1	BT Transmission On-going
coex_bt_rx	input	1	BT ReceptionOn-going
coex_bt_event	input	1	BT EventOn-going
coex_bt_tx_abort	output	1	BT Transmission Abort Request
coex_bt_rx_abort	output	1	BT ReceptionAbort Request
coex_bt_pti	input	4	BT Packet Traffic Information
coex_bt_channel	input	7	BT Channel (0-78)
coex_bt_bw	input	1	BT Bandwidth (0:1MHz, 1:2MHz)
RC Coexistence interface			
rc_tx_abort	output	1	Radio Controller Transmission Abort Request
rc_rx_abort	output	1	Radio Controller ReceptionAbort Request
AXI Interface			
upstream interface			
dma0_awid	output	4	DMA Upstream awid
dma0_awaddr	output	32	DMA Upstream awaddr
dma0_awlen	output	8	DMA Upstream awlen
dma0_awsize	output	3	DMA Upstream awsize
dma0_awburst	output	2	DMA Upstream awburst
dma0_awuser	output	12	DMA Upstream awuser
dma0_awvalid	output	1	DMA Upstream awvalid
dma0_awready	input	1	DMA Upstream awready
dma0_wid	output	4	DMA Upstream wid
dma0_wdata	output	64	DMA Upstream wdata
dma0_wstrb	output	8	DMA Upstream wstrb
dma0_wlast	output	1	DMA Upstream wlast
dma0_wvalid	output	1	DMA Upstream wvalid
dma0_wready	input	1	DMA Upstream wready
dma0_bid	input	4	DMA Upstream bid
dma0_bresp	input	2	DMA Upstream bresp
dma0_bvalid	input	1	DMA Upstream bvalid
dma0_bready	output	1	DMA Upstream bready
downstream interface			
dma1_arid	output	4	DMA Downstream arid
dma1_araddr	output	32	DMA Downstream araddr
dma1_arlen	output	8	DMA Downstream arlen
dma1_arsize	output	3	DMA Downstream arsize
dma1_arburst	output	2	DMA Downstream arburst
dma1_arvalid	output	1	DMA Downstream arvalid
dma1_arready	input	1	DMA Downstream arready
dma1_rid	input	4	DMA Downstream rid
dma1_rdata	input	64	DMA Downstream rdata
dma1_rresp	input	2	DMA Downstream rresp
dma1_rlast	input	1	DMA Downstream rlast
dma1_rvalid	input	1	DMA Downstream rvalid
dma1_rready	output	1	DMA Downstream rready
ADC Interface			

adc0_on	output	1	ADC0 Power/Clock enable
adc0_i	input	12	ADC0 Data I
adc0_q	input	12	ADC0 Data Q
adc1_on	output	1	ADC1 Power/Clock enable
adc1_i	input	12	ADC1 Data I
adc1_q	input	12	ADC1 Data Q
DAC Interface			
dac0_on	output	1	DAC0 Power/Clock enable
dac0_en	output	1	DAC0 Data valid
dac0_i	output	12	DAC0 Data I
dac0_q	output	12	DAC0 Data Q
dac1_on	output	1	DAC1 Power/Clock enable
dac1_en	output	1	DAC1 Data valid
dac1_i	output	12	DAC1 Data I
dac1_q	output	12	DAC1 Data Q
RF interface			
Reset output			
rf_resetn	output	1	Active Low Reset for Radio
RF AGC Control			
rf_agcfreeze	output	1	AGC freeze
RF Test mode			
rf_tmode	output	1	Radio Test Mode
Fast write bus			
rf_gpio	output	8	GPIO Data
SPI Interface			
rf_spi_in	input	1	SPI Interface In
rf_spi_ss_n	output	1	SPI Interface Slave Select
rf_spi_clk	output	1	SPI Interface Clock
rf_spi_out	output	1	SPI Interface Out
Tx/Rx switch			
rf_trsw0	output	1	Radio Tx On 0
rf_trsw1	output	1	Radio Tx On 1
External PA			
rf_extpaon_ch0_5G9	output	1	Radio External PA Channel 0 5Ghz
rf_extpaon_ch0_2G4	output	1	Radio External PA Channel 0 2.4Ghz
rf_extpaon_ch1_5G9	output	1	Radio External PA Channel 1 5Ghz
rf_extpaon_ch1_2G4	output	1	Radio External PA Channel 1 2.4Ghz
AHB to FPGA			
fpga_hready_in	output	1	FPGA hready_in
fpga_hsel	output	1	FPGA hsel
fpga_haddr	output	20	FPGA haddr
fpga_htrans	output	2	FPGA htrans
fpga_hwrite	output	1	FPGA hwrite
fpga_hrdata	input	32	FPGA hrdata
fpga_hwdata	output	32	FPGA hwdata
fpga_hresp	input	2	FPGA hresp
fpga_hready	input	1	FPGA hready
Diagnostic Ports			
ext_diagport	input	32	Diagnostic port
reg_bootrom_enable	output	1	CPU Boot enable
rw_nx_ss_diag	output	5	Diagnostic port to be used with IQ for source synchrone capture
rw_nx_diag0	output	32	Diagnostic port
rw_nx_diag1	output	32	Diagnostic port
rw_nx_diag2	output	32	Diagnostic port
mac_internal_error	output	1	MAC internal Error

mac_debug_ksr	input	1	Debug Key Storage RAM.
macbypass_trigger	output	2	MAC Bypass Trigger Port
reg_diag_trigger	output	1	SW Trigger Port
Diagnostic Ports			
jtag_rtck	output	1	output wire jtag_rtck,
jtag_tdo	output	1	output wire jtag_tdo,
jtag_tdi	input	1	inputwire jtag_tdi,
jtag_tms	input	1	inputwire jtag_tms,
jtag_tck	input	1	inputwire jtag_tck

Table 4.3: Other signals

5 Implementation

5.1 Area report

5.1.1 Gate count

The typical area report and gate count of RW-WLAN nX IP including RISV CPU in several configurations using TSMC 40nm (LP) is shown in table below:

Configuration	CHBW support	Gate Count total (K Gates)
1x1	20 MHz	1370
1x1 LDPC	20 MHz	1716
1x1 LDPC	20/40 MHz	1854

Table 5-1: gate count in nand2x1 in TSMC 40nm (LP)

5.2 Clock tree

The RW-WLAN HE TOP IP has been designed to insert easily the clock tree. All the clock sources are in the rw_he_crm block provided as example for the integrator. The RW-WLAN HE MAC and Modem contains several clocks gathered into 5 groups that they can be balanced independently: one for PHY and MAC/PHY, one for FE and AGC, one for DSSS-CCK, one for RxBD, one for the AHB clock domain and one for the MAC.

5.3 Reset tree

The strategy for reset tree in the RW-WLAN nX IP is to synchronously remove the reset on a flip-flop pin with the clock clocking this same flip-flop.

5.4 Power consumption

The table below contains the power consumption results of the RW-WLAN HE Modem using TSMC 90nm (LP).

Operating condition	Results TSMC 90nm
802.11b Tx 11Mbps	TBD-2
802.11b Rx 11Mbps	TBD-2
802.11a Tx 54Mbps	TBD-2
802.11a Rx 54Mbps	TBD-2
802.11n Tx MCS7	TBD-2
802.11n Rx MCS7	TBD-2
802.11n Tx MCS15	TBD-2
802.11n Rx MCS15	TBD-2
802.11ac Tx MCS9 1ss	TBD-2
802.11ac Rx MCS9 1ss	TBD-2
802.11ac Tx MCS9 2ss	TBD-2
802.11ac Rx MCS9 2ss	TBD-2

Table 5-2: Power consumption results

5.5 DFT

The RW-WLAN HE IP is designed for use with classic scan test.

During test mode, the reset controller functionality must be bypassed, so that test data coming on the internal reset source inputs (for instance) do not reset the IP.

5.6 Synthesis flow

Logic synthesis is conducted with the Synopsys DC-Ultra synthesis tool. The set of BASH/TCL scripts detailed in Table 5-3 is supplied:

Script file	Description
asic_run.sh	front-end script that prepare/run the synthesis : build the synthesis workarea collect list of files to be synthesized launch dc_shell
rw_he_top_ cpu_commands.tcl	Analyzes RTL sources Applies constraints Synthesizes design Produces intermediary and final reports and databases
rw_he_top_cpu_constraints.tcl	Defines timing constraints
rw_he_top_ cpu_clockdef.tcl	Defines clocking system
library_setup.tcl	Sets custom control variables Loads technology files

Table 5-3: Synthesis scripts set

These files are located in <…>/IPs/HW/TOP11AX/rw_he_top_cpu/synth/ASIC/_dc_template/scripts and <…>/env/SYNTH directories.

5.6.1 How to synthesize the RW-WLAN HE TOP CPU

1) Ensure that appropriate environment variables about the project and tools are initialized.
2) Edit paths within the library_setup.tcl in order to point to your target library.
3) Change into the IP synthesis directory:

   'cd $SOURCESLIB/IPs/HW/ TOP11AX/rw_he_top_cpu/synth/ASIC'

4) Build a work area by running the asic.sh frontend script for a given configuration:
   a. For instance for a 1x1-20 MHz:
      'RW_SYNDIR=dev ./asic.sh –i -c –config CONFIG_STA_1x1_CBW20'

   b. For instance for a 1x1-40MHz with LDPC:
      'RW_SYNDIR=dev ./asic.sh –i –c –config CONFIG_STA_1x1_CBW40_LDPC'

5) Launch the synthesis with the desired option:
   a. For instance for a 1x1-20 MHz:
      'RW_SYNDIR=dev ./asic.sh –config CONFIG_STA_1x1_CBW20 –s'

   b. For instance for a 1x1-40MHz with LDPC:
      'RW_SYNDIR=dev ./asic.sh –config CONFIG_STA_1x1_CBW40_LDPC –s'

References

[1] RW-WLAN-nX Modem Functional Specification, RW-WLAN-nX-Modem-FS, RivieraWaves.

[2] RW-WLAN-nX-Modem User Manual, RW-WLAN-nX-Modem-UM, RivieraWaves

[3] RW-WLAN-nX Modem Functional Specification, RW-WLAN-nX-MAC-HW-STA-FS, RivieraWaves.

[4] RW-WLAN-nX-Modem User Manual, RW-WLAN-nX-MAC-HW-STA-UM, RivieraWaves

[5] RW-WLAN-nX-Modem User Manual, RW-WLAN-HE-TOP-HW-IG, RivieraWaves