CinvestavDupin - RISC-V Microcontroller

Diego H - FPGAParadox - DRL

Contents

Release Notes

- Fixed #1 "Senal START Interfaz"
- Fixed #3 The data_out of dummy interface IP is held for more than one clock cycle
  - From the following figure, it can be seen that no **read** is issued without a **data_out** output.

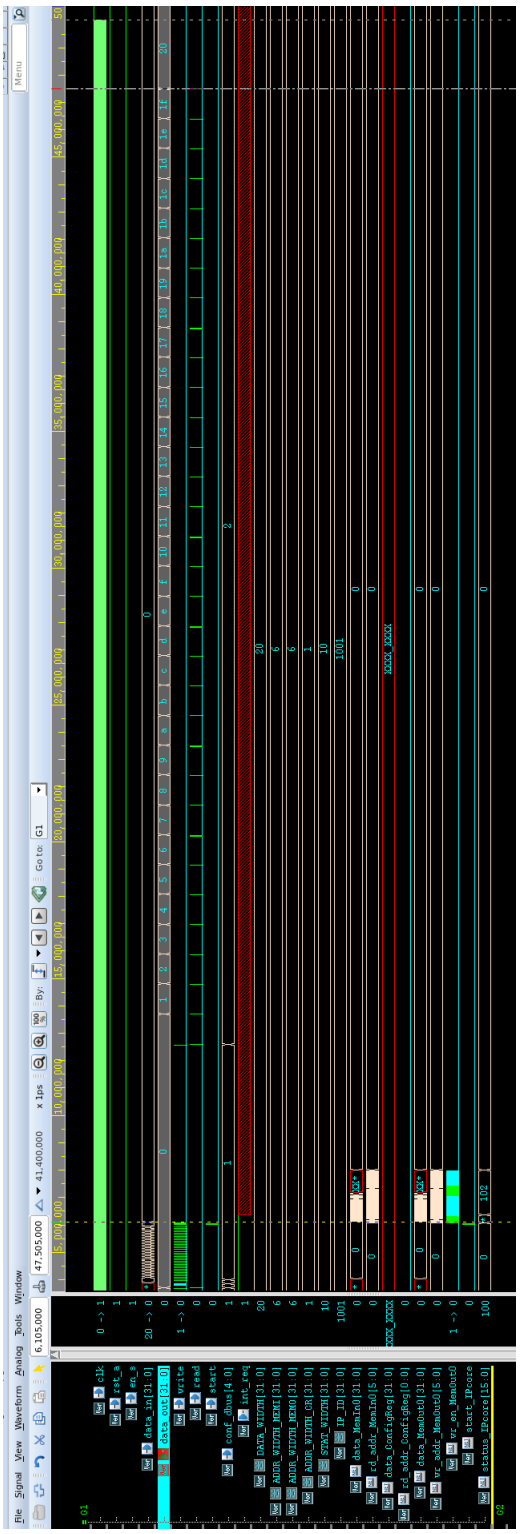

Figure 1: Fixed #3.

- Released CinvestavDupin following requirements specification sheet.

Part I
Architectural Overview

1 Requirements

The design requirements that **CinvestavDupin** meets are listed below.

- General I/O:
  - Reset
  - Clock
  - Interrupts x4
  - JTAG
- Low Speed I/O:
  - GPIO x32
  - Timer
  - SPI
  - UART
  - I2C
  - **AIP NoC**

All of the above Low Speed I/Os comply with AMBA protocol. AIP is the only exception as this is an external IP of Dupin and Titan IP catalog.

2 Bird's-Eye View

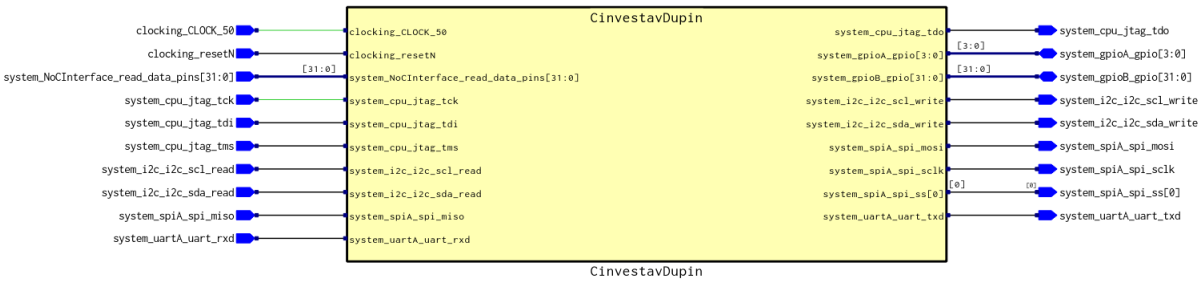

Figure 2: CinvestavDupin Block View.

3 Resource Utilisation

The resource utilisation of Dupin in the CinvestavDupin SoC is shown below. Dupin is configured to use as small area as possible, without compromising functionality or speed.

This numbers are by using smaller Intel Cyclone IV device. The FMax reported by Synplify is 61.5 MHz, altough **the main clock frequency is set to 50MHz**. [1].

Module name	ATOMS	ARITHMETIC M	REGISTERS	SYNC RAMS	MACs
top	2214	0	1575	8	0
CinvestavDupin	2213	0	1575	8	0
Apb3Decoder	16	0	0	0	0
Apb3Gpio2	41	0	36	0	0
Apb3Gpio2_1	60	0	128	0	0
Apb3I2cCtrl	286	0	204	0	0
Apb3Router	133	0	3	0	0
Apb3SpiMasterCtrl	172	0	205	2	0
Apb3UartCtrl	124	0	81	0	0
BufferCC_6	2	0	2	0	0
BufferCC_7	0	0	2	0	0
DupinCore	1033	0	418	2	0
JtagBridge	35	0	122	0	0
MachineTimer	23	0	129	0	0
SystemDebugger	15	0	78	0	0
hookIFArbiter	51	0	0	0	0
hookIFDecoder	37	0	6	0	0
hookIFDecoder_1	107	0	8	0	0
hookIFOnChipRam	3	0	3	4	0
hookIFToApb3Bridge	1	0	35	0	0

Figure 3: CinvestavDupin Resource Utilisation.

Part II
Software View

4 Address Map

The base address map of the SoC is shown below. For the sake of clarity, all IPs behind APB router uses APB Router base address + component base address for the final address calculation. For instance, UART final base address is 0x10010000.

[1] For the sake of comparison, Virtex 7 smaller device reports 280MHz, whereas Cyclone V largest device reports 119.8MHz.

Device	Address
On-chip RAM	0x80000000
UART	0x10000
Interrupts (GPIO A)	0x00000
GPIO B	0x50000
Timer	0x08000
I2C	0x60000
PLIC	**0xC00000**
SPI	0x70000
NoC	0x40000
APB Router	0x10000000

Table 1: DupinSoC Address Map.

5 Board Support Package (BSP)

Most drivers are provided under $DUPIN_REPO/bsp, including a DTS, since SiFive's new FreedomStudio requieres a DTS file for debugging purposes.

6 Platform-Level Interrupt Controller (PLIC)

The RISC-V architecture uses the privileged mode to service both internal and external interrupts. For external interrupts, the PLIC manages the interrupt sources as well as the priorities and settings.

6.1 High Level PLIC Configuration

The steps to configure and use PLIC are depicted below:

- Configure the PLIC to accept interrupts above priority 0.
- Enable the interrupt I/O.
- Set a priority for the interrupt I/O.
- Set the interrupt level/edge settings.
- Apply the configuration to the SoC using the CSRs.
  - Set MTVEC register with the trap subroutine, defined in your startup.S file. For Dupin, is called **trap_entry**.
  - Set MIE register with the interrupt source.
  - Set MSTATUS register with the interrupt source.

For usage, follow this steps:

- When an interrupt is present, the **trap_entry** calls **trap()** function in your main.
  - The **trap()** must get the MCAUSE and mask the interrupt (discriminate exceptions).

- Within **trap()** and after masking MCAUSE, the interrupt cause is obtained. Either the external interrupt subroutine or timer subroutine can be called from here.
- The software developer must mask the claimed interrupt, and call the subroutine accordingly.
- Finally, claim must be cleared (unmasked) from PLIC.

This is standard work flow of RISC-V interrupts and has nothing to do with Dupin architectural conception.

An example is provided in **$DUPIN_REPO/sw/dummy_interrupts/src/main.c**