

---

# **Raisin64 Documentation**

***Release 0.1***

**Christopher Parish**

**Oct 10, 2018**

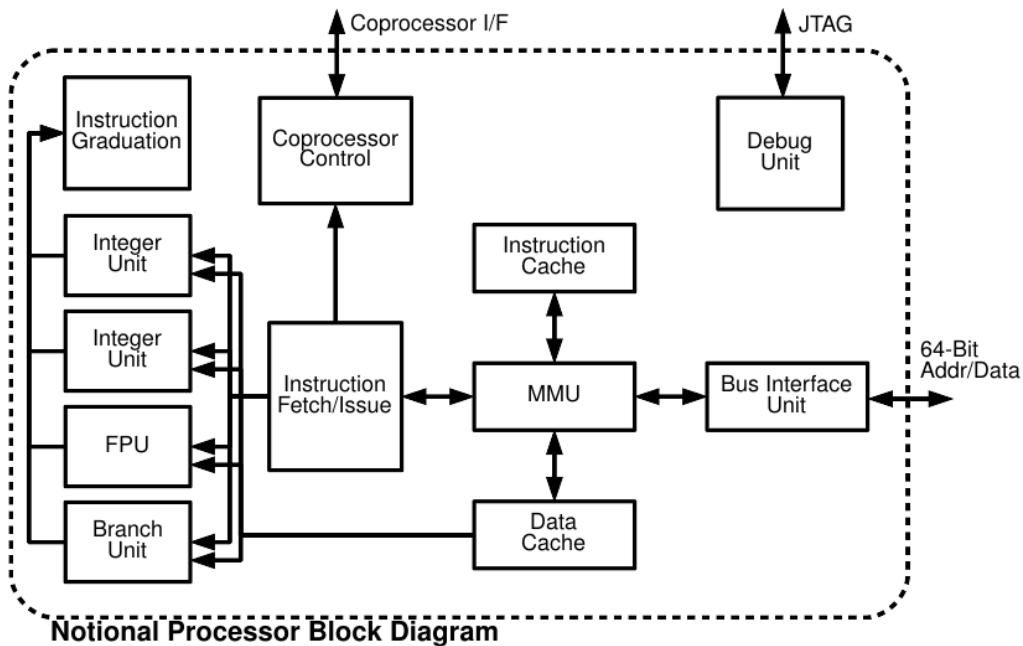


**CONTENTS:**

<b>1</b>	<b>Raisin64 CPU</b>	<b>3</b>
1.1	Overview . . . . .	3
1.2	Pipeline Stages . . . . .	3
1.3	Caches . . . . .	3
1.4	MMU . . . . .	3
1.5	Interrupt Unit . . . . .	3
1.6	Debug Unit . . . . .	3
<b>2</b>	<b>Code Snippets and Software</b>	<b>5</b>
2.1	Handling Interrupts . . . . .	5
2.2	Initializing the MMU . . . . .	5
<b>3</b>	<b>Tools</b>	<b>7</b>
3.1	Assembler . . . . .	7
3.2	Debugging . . . . .	7
<b>4</b>	<b>Nexys 4 DDR Reference Implementation</b>	<b>9</b>
4.1	SoC Peripherals . . . . .	9
4.2	Required Hardware . . . . .	9
4.3	Synthesizing the Core . . . . .	9
<b>5</b>	<b>Source Index</b>	<b>11</b>
5.1	Verilog Module Index . . . . .	11



Raisin64 (*RISC Architecture with In-order Superscalar INterlocked-pipeline*) is a pure 64-bit CPU design created as part of an educational project. Architecturally similar to the [MIPS R10000](#) and [POWER3](#), Raisin64 is a superscalar design that employs multiple specialized pipelines for integer operations, floating point, load/store, etc. Unlike most superscalar designs, Raisin64 does not re-order instructions but instead provides a larger architectural register file of 64x64-bit registers.



Major features of the Raisin64 include:

- **Bits:** 64-bit
- **Design:** RISC
- **Type:** Register-Register
- **Branching:** Condition Code
- **Endianness:** Big
- **Page Size:** 16KB Fixed
- **Virtual Address Size:** 47-Bits
- **Page Table:** Three Level
- **Registers:** 61 (R0 = 0)



**RAISIN64 CPU**

**1.1 Overview**

**1.2 Pipeline Stages**

**1.3 Caches**

**1.4 MMU**

**1.5 Interrupt Unit**

**1.6 Debug Unit**





## CODE SNIPPETS AND SOFTWARE

### 2.1 Handling Interrupts

### 2.2 Initializing the MMU



- *Assembler*
- *Debugging*
  - *Getting OpenOCD*

## 3.1 Assembler

## 3.2 Debugging

### 3.2.1 Getting OpenOCD



## **NEXYS 4 DDR REFERENCE IMPLEMENTATION**

### **4.1 SoC Peripherals**

### **4.2 Required Hardware**

### **4.3 Synthesizing the Core**



## SOURCE INDEX

### 5.1 Verilog Module Index

#### 5.1.1 Raisin64.v

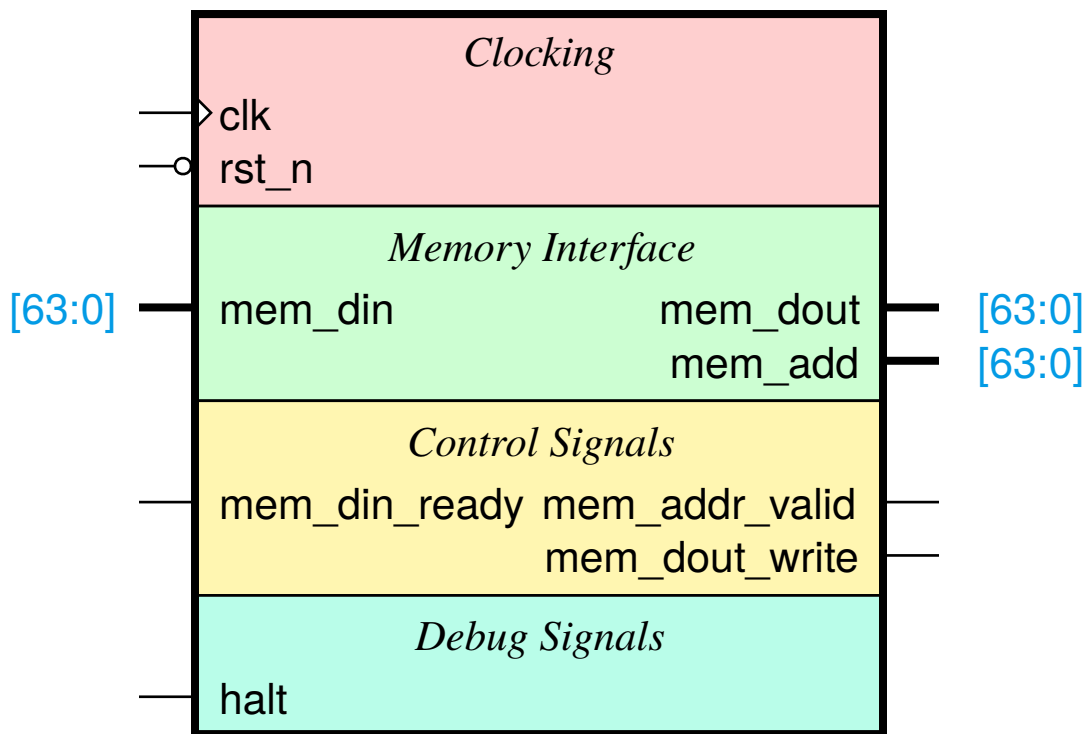


Fig. 1: Raisin64.v

```
1  /*  
2   * Raisin64 CPU  
3   */  
4  
5  module raisin64 (  
6      //# {{clocks|Clocking}}  
7      input clk,  
8      input rst_n,
```

(continues on next page)

(continued from previous page)

```
9
10  //# {{data|Memory Interface}}
11  input [63:0] mem_din,
12  output [63:0] mem_dout,
13  output [63:0] mem_add,
14
15  //# {{control|Control Signals}}
16  output mem_addr_valid,
17  output mem_dout_write
18  input mem_din_ready,
19
20  //# {{debug|Debug Signals}}
21  input halt);
22
23 endmodule
```