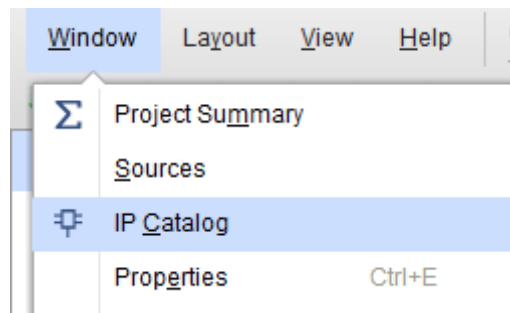


How To Change the Frequency of your CPU

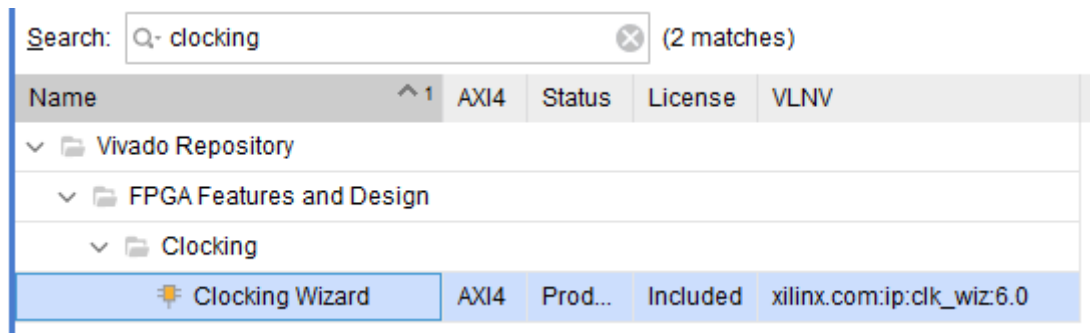
You may have heard of 'overclock', which can squeeze out more performance of your Intel or AMD CPU. We can do the same to our own CPU by changing the frequency of the clock.

You need to modify some parameters in riscv_top.v and add a new Clock_wiz module generated automatically by Vivado.

1. Choose IP Catalog



2. Find Clock Wizard



3. Choose the frequency you want to change to, 150mhz as an example

Clocking Wizard (6.0)

Documentation IP Location Switch to Defaults

IP Symbol Resource

☒ Show disabled ports

Component Name: clk_wiz_0

Clocking Options **Output Clocks** Port Renaming MMCM Settings Summary

The phase is calculated relative to the active input clock.

Output Clock	Port Name	Output Freq (MHz)		Phase (degrees)		Duty Cycle (%)		Drives
		Requested	Actual	Requested	Actual	Requested	Actual	
<input checked="" type="checkbox"/> clk_out1	clk_out1	150.000	150.000	0.000	0.000	50.000	50.0	BUFG
<input type="checkbox"/> clk_out2	clk_out2	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
<input type="checkbox"/> clk_out3	clk_out3	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
<input type="checkbox"/> clk_out4	clk_out4	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
<input type="checkbox"/> clk_out5	clk_out5	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
<input type="checkbox"/> clk_out6	clk_out6	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
<input type="checkbox"/> clk_out7	clk_out7	100.000	N/A	0.000	N/A	50.000	N/A	BUFG

☐ USE CLOCK SEQUENCING

Clocking Feedback

Output Clock	Sequence Number
clk_out1	1
clk_out2	1
clk_out3	1
clk_out4	1
clk_out5	1
clk_out6	1
clk_out7	1

Source

☒ Automatic Control On-Chip
☐ Automatic Control Off-Chip
☐ User-Controlled On-Chip
☐ User-Controlled Off-Chip

Signaling

☒ Single-ended
☐ Differential

Enable Optional Inputs / Outputs for MMCM/PLL Reset Type

OK Cancel

Then press OK and generate output products

The following output products will be generated.

Preview

clk_wiz_0.xci (OOC per IP)

- Instantiation Template
- Synthesized Checkpoint (.dcp)
- Structural Simulation
- Change Log

Synthesis Options

☐ Global
☒ Out of context per IP

Run Settings

Number of jobs: 6

4. Add clkwizard module in riscv_top.v

Once Clk_wiz module is generated, you should add an instance of this module in riscv_top.v

```

localparam SYS_CLK_FREQ = 150000000;
localparam UART_BAUD_RATE = 115200;
localparam RAM_ADDR_WIDTH = 17;           // 128KiB

reg rst;
reg rst_delay;

wire clk;

// assign EXCLK (or your own clock module) to clk
// assign clk = EXCLK;
wire locked;
clk_wiz_0 NEW_CLOCK(
    .reset(btnC),
    .clk_in1(EXCLK),
    .clk_out1(clk),
    .locked(locked)
);

```

Remember to modify the SYS_CLK_FREQ the same as the frequency in Clk_wizard module

5. Generate bitstream as normal

Implementation	Summary Route Status
Status:	✓ write_bitstream Complete!
Messages:	<div> <div>⚠ 1 critical warning</div> <div>⚠ 2 warnings</div> </div>
Part:	xc7a35tcpg236-1
Strategy:	Vivado Implementation Defaults
Report Strategy:	Vivado Implementation Default Reports
Incremental implementation:	None
Timing	Setup Hold Pulse Width
Worst Negative Slack (WNS):	-3.463 ns
Total Negative Slack (TNS):	-8988.208 ns

You may find the timing is negative and thus leads to a critical warning, but sometimes you can still run your CPU on FPGA, so just try.

6. Improvement as an example

```

CPU start
314159265358979323846264338327952884197169399375105820974944592307816406286208998628034825342117067982148086513282366470
938446955582231725359408128481117450284127019385211555964462294895493038196442881097566593344612847564823378678316527120
199145648566923463486104543266482133936072602491412737245870660631558817488152092096282925491715364367892590361133053054
882046652138414695194151160943305727365759591953092186117381932611793105118548074462379962749567351885752724891227938183
011949129833673362440656643860213949463952247371907021798694370277053921717629317675238467481846766945132000568127145263
560827785771342757789609173637178721468440901224953430146549585371057922796892589235420199561121290219686434418159813629
7747713099605187072113499999983729780499510597317328160963185
CPU returned with running time: 3.593750

```

pi at 100 mhz

```
└─
CPU start
314159265358979323846264338327952884197169399375105820974944592307816406286208998628034825342117067982148086513282366470
938446955582231725359408128481117450284127019385211555964462294895493038196442881097566593344612847564823378678316527120
199145648566923463486104543266482133936072602491412737245870660631558817488152092096282925491715364367892590361133053054
882046652138414695194151160943305727365759591953092186117381932611793105118548074462379962749567351885752724891227938183
011949129833673362440656643860213949463952247371907021798694370277053921717629317675238467481846766945132000568127145263
560827785771342757789609173637178721468440901224953430146549585371057922796892589235420199561121290219686434418159813629
7747713099605187072113499999983729780499510597317328160963185
CPU returned with running time: 1.234375
```

pi at 200 mhz