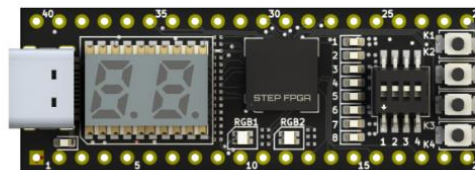
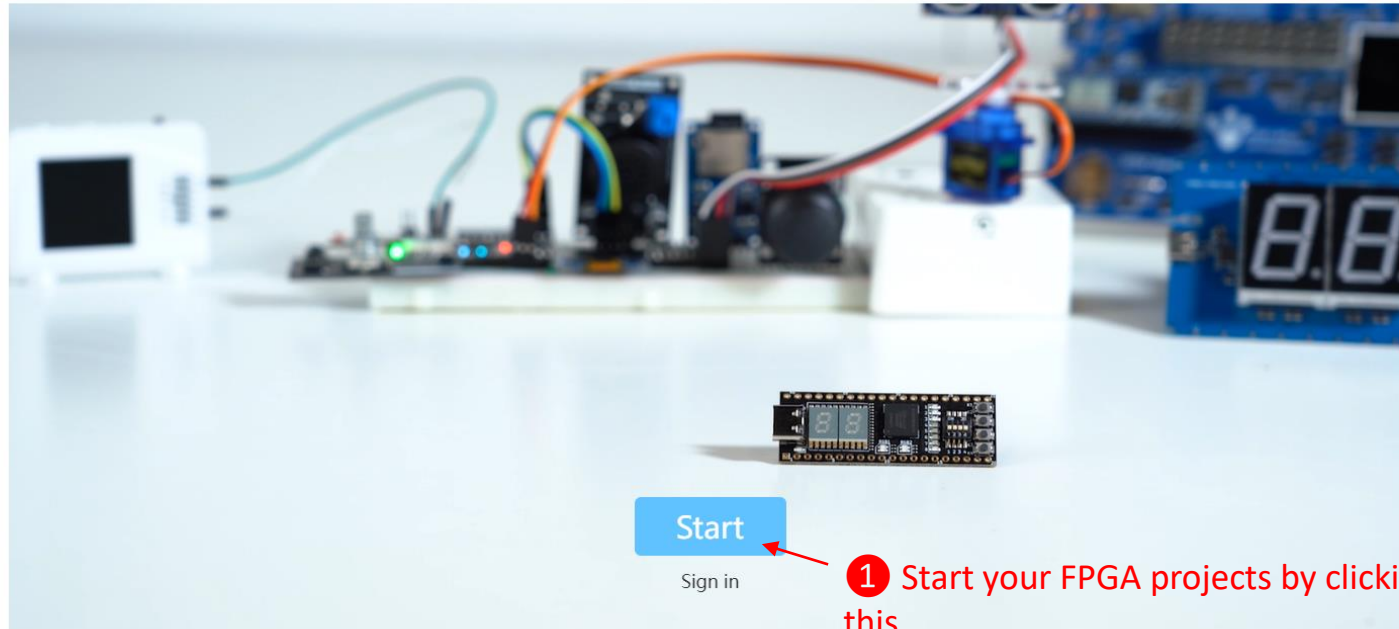
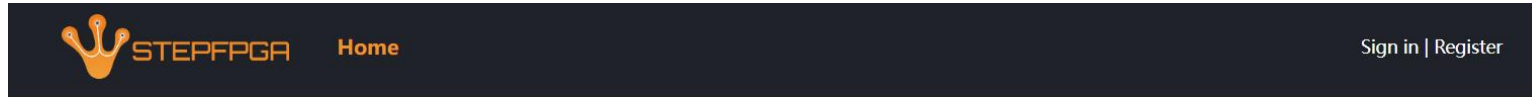


# Start your FPGA project on WebIDE (stepfpga.eimtechnology.com)



Design, Develop, and Share your FPGA source code

# Create a project

The screenshot shows the 'Create' project page on the STEP FPGA website. The interface includes a top navigation bar with the STEP FPGA logo and 'Home' link, and a secondary bar with icons for 'Source code', 'Logic synthesis', 'Pin assignment', 'FPGA mapping', 'Download', 'Example project', and 'Instructions'. A left sidebar contains 'New', 'Documents', 'Logs', and 'Result' options. The main 'Create' form has the following fields and annotations:

- Project name:** An input field with a red arrow pointing to it and the annotation "1 Name your project first". Below it is a warning icon and the text "Name of your Verilog file, begin with Letters (max 100 chars)".
- Board:** A dropdown menu with "Please select the board" as the current selection. The dropdown is open, showing "STEP-MXO2 V2.2" and "STEP-MXO2Core" as options. A red arrow points to "STEP-MXO2 V2.2" with the annotation "2 Retiring version; MicroUSB connector". Another red arrow points to "STEP-MXO2Core" with the annotation "3 New version; Type-C connector".
- Project tag:** An input field with the placeholder text "Such as: combinational logic, sequential logic". Below it is a warning icon and the text "Optional: add a name tag to categorize your project".
- Description:** A large text area for project details. A red arrow points to it with the annotation "4 Project tag and descriptions are optional and for your own references".
- Submit:** An orange button at the bottom of the form. A red arrow points to it with the annotation "5 Click this to finish creating your project".

# Create a Verilog design file

**STEPFPGA** Home

decoder38

Source code Logic synthesis Pin assignment FPGA mapping Download Example project Instructions

Documents

Logs

Result

1 Click to Add new Verilog design files under this project

2 Click this one

Create a new file

Choose from project

# Code Editing Interface

The screenshot displays the STEPFPGA web interface. At the top left is the STEPFPGA logo and the word "Home". Below this is a navigation bar with icons for "Source code", "Logic synthesis", "Pin assignment", "FPGA mapping", "Download", "Example project", and "Instructions". On the left side, there is a sidebar with "Documents" containing "decoder38.v", "Logs", and "Result". The main area is a code editor for "decoder38.v" showing Verilog code. A red arrow labeled "1" points to the code editor with the text "Design your Verilog code here". On the right side, there is a button that says "Click Save (Ctrl/Command + S)". A red arrow labeled "2" points to this button with the text "Click Save after editing".

STEPFPGA Home

decoder38 ▾

Source code Logic synthesis Pin assignment FPGA mapping Download Example project Instructions

Documents

decoder38.v

Logs

Result

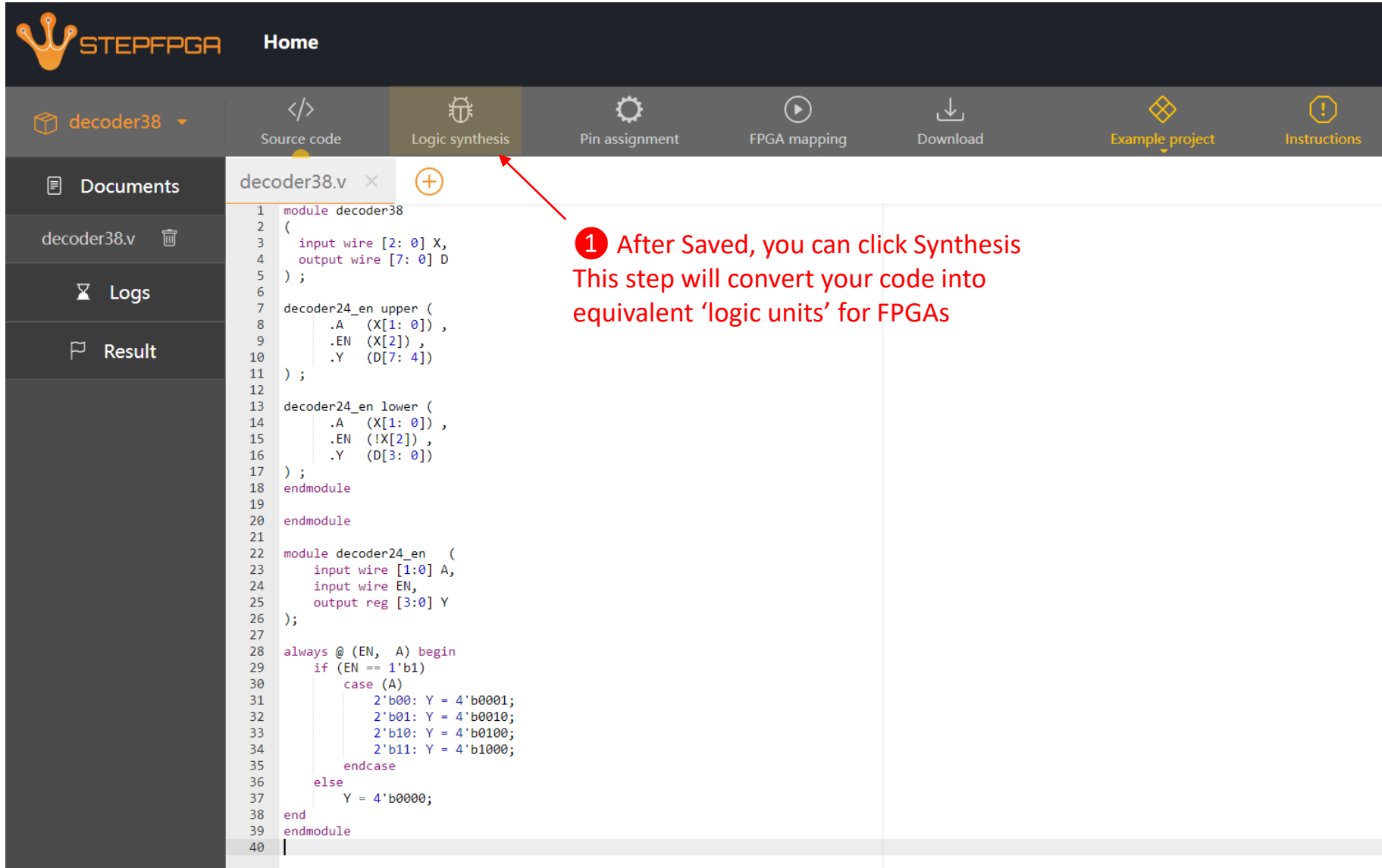
```
1 module decoder38 (  
2     input [2:0] A,  
3     output [7:0] Y  
4 );  
5  
6  
7 // ...  
8 // ...  
9  
10  
11 endmodule  
12  
13 |
```

Click Save (Ctrl/Command + S)

1 Design your Verilog code here

2 Click Save after editing

## Continue to Logic Synthesis



The screenshot displays the STEPFGA web interface. At the top, the logo and 'Home' are visible. Below is a navigation bar with icons for 'decoder38', 'Source code', 'Logic synthesis', 'Pin assignment', 'FPGA mapping', 'Download', 'Example project', and 'Instructions'. The 'Logic synthesis' icon is highlighted with a red arrow. On the left, a sidebar shows 'Documents' with 'decoder38.v', 'Logs', and 'Result'. The main area shows a code editor for 'decoder38.v' with the following code:

```
1 module decoder38
2 (
3   input wire [2: 0] X,
4   output wire [7: 0] D
5 );
6
7 decoder24_en upper (
8   .A (X[1: 0]) ,
9   .EN (X[2]),
10  .Y (D[7: 4])
11 );
12
13 decoder24_en lower (
14   .A (X[1: 0]) ,
15   .EN (!X[2]),
16   .Y (D[3: 0])
17 );
18 endmodule
19
20 endmodule
21
22 module decoder24_en (
23   input wire [1:0] A,
24   input wire EN,
25   output reg [3:0] Y
26 );
27
28 always @ (EN, A) begin
29   if (EN == 1'b1)
30     case (A)
31       2'b00: Y = 4'b0001;
32       2'b01: Y = 4'b0010;
33       2'b10: Y = 4'b0100;
34       2'b11: Y = 4'b1000;
35     endcase
36   else
37     Y = 4'b0000;
38 end
39 endmodule
40
```

A red circle with the number '1' is placed over the 'Logic synthesis' icon, with a red arrow pointing to it. To the right of the icon, the following text is written in red:

1 After Saved, you can click Synthesis  
This step will convert your code into  
equivalent 'logic units' for FPGAs

## A tip for quick design

STEPFPGA Home

decoder38 ▾

Source code Logic synthesis Pin assignment FPGA mapping Download Example project Instructions

Documents

decoder38.v

Logs

Result

```
1 module decoder38
2 (
3   input wire [2: 0] X,
4   output wire [7: 0] D
5 );
6
7 decoder24_en upper (
8   .A (X[1: 0]) ,
9   .EN (X[2]) ,
10  .Y (D[7: 4])
11 );
12
13 decoder24_en lower (
14   .A (X[1: 0]) ,
15   .EN (!X[2]) ,
16   .Y (D[3: 0])
17 );
18 endmodule
19
20 module decoder24_en (
21   input wire [1:0] A,
22   input wire EN,
23   output reg [3:0] Y
24 );
25
26 always @ (EN, A) begin
27   if (EN == 1'b1)
28     case (A)
29       2'b00: Y = 4'b0001;
30       2'b01: Y = 4'b0010;
31       2'b10: Y = 4'b0100;
32       2'b11: Y = 4'b1000;
33     endcase
34   else
35     Y = 4'b0000;
36 end
37 endmodule
38
```

1 Your assistant for often-used code managements

# Manage your own workpieces

The screenshot displays the STEPPFGA WebIDE interface. At the top, there is a navigation bar with icons for Source code, Logic synthesis, Pin assignment, FPGA mapping, Download, Example project, and Instructions. Below this is a sidebar with 'Documents' and 'Result' sections. The main area is a code editor showing Verilog code for a decoder38 module. A sidebar on the right contains a list of modules, with 'Common' and 'Custom' tabs at the bottom. Red annotations highlight the 'Custom' tab and a specific module in the list.

```
1 module decoder38
2 (
3   input wire [2: 0] X,
4   output wire [7: 0] D
5 );
6
7 decoder24_en upper (
8   .A (X[1: 0]),
9   .EN (X[2]),
10  .Y (D[7: 4])
11 );
12
13 decoder24_en lower (
14   .A (X[1: 0]),
15   .EN (X[2]),
16   .Y (D[3: 0])
17 );
18 endmodule
19
20 module decoder24_en (
21   input wire [1:0] A,
22   input wire EN,
23   output reg [3:0] Y
24 );
25
26 always @ (EN, A) begin
27   if (EN == 1'b1)
28     case (A)
29       2'b00: Y = 4'b0001;
30       2'b01: Y = 4'b0010;
31       2'b10: Y = 4'b0100;
32       2'b11: Y = 4'b1000;
33     endcase
34   else
35     Y = 4'b0000;
36 end
37 endmodule
38
```

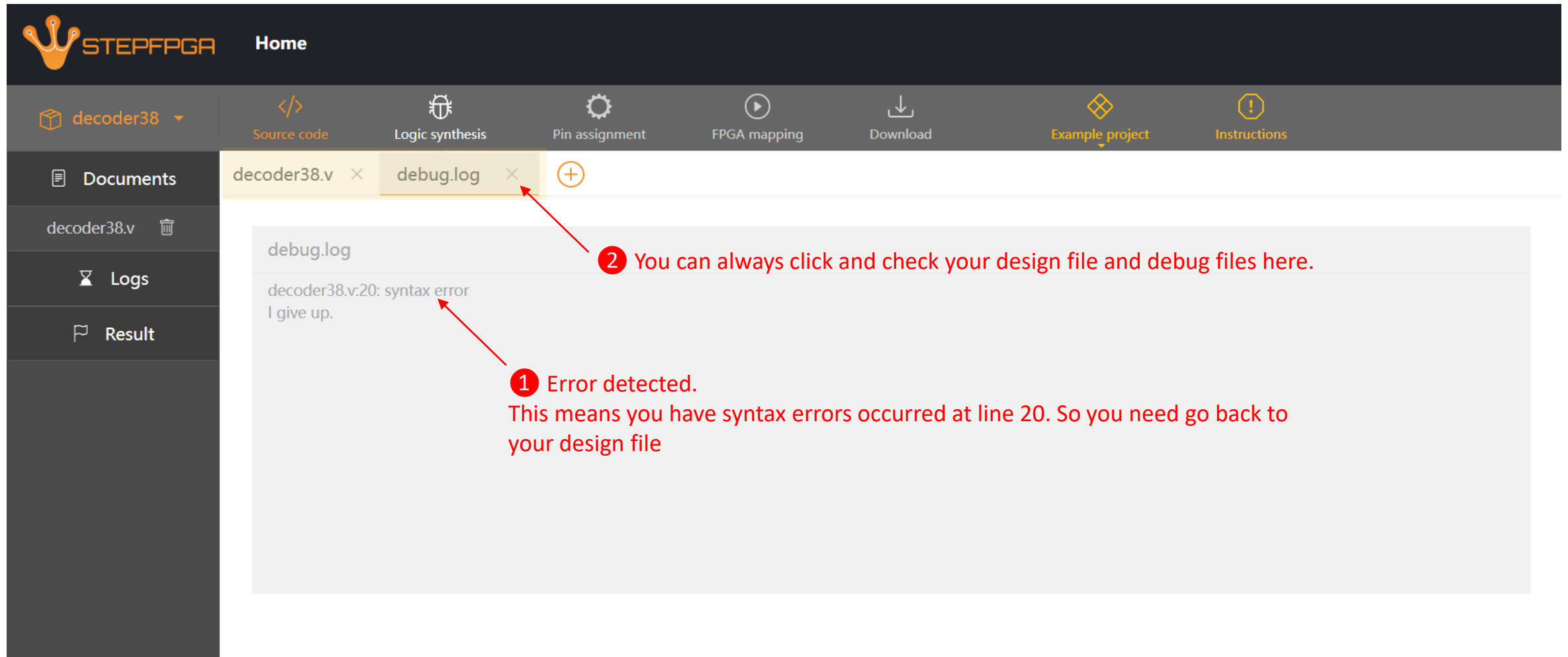
**1** The WebIDE will keep update some commonly used modules for your quick reference

**2** You can also save your own favorite modules for future usage

gate_and	gate_or
gate_not	gate_xor
gate_nand	gate_nor
adder_half	adder_full
decoder24	decoder38
encoder42	encoder83
mux2	mux4
flipflop_d	
reg4_shift	
clock_divider_integer	
counter8_ring	
counter8_twist	
pwm50Hz_var duty	
pwm1kHz_var duty	
pwm10kHz_var duty	

Common Custom

## Showing debug information of your code



The screenshot shows the STEPFPGA web interface. At the top left is the logo and the text "STEPFPGA Home". Below this is a navigation bar with icons for "Source code", "Logic synthesis", "Pin assignment", "FPGA mapping", "Download", "Example project", and "Instructions". On the left side, there is a sidebar with "Documents" containing "decoder38.v", "Logs", and "Result". The main area shows two tabs: "decoder38.v" and "debug.log". The "debug.log" tab is active and displays the following text:

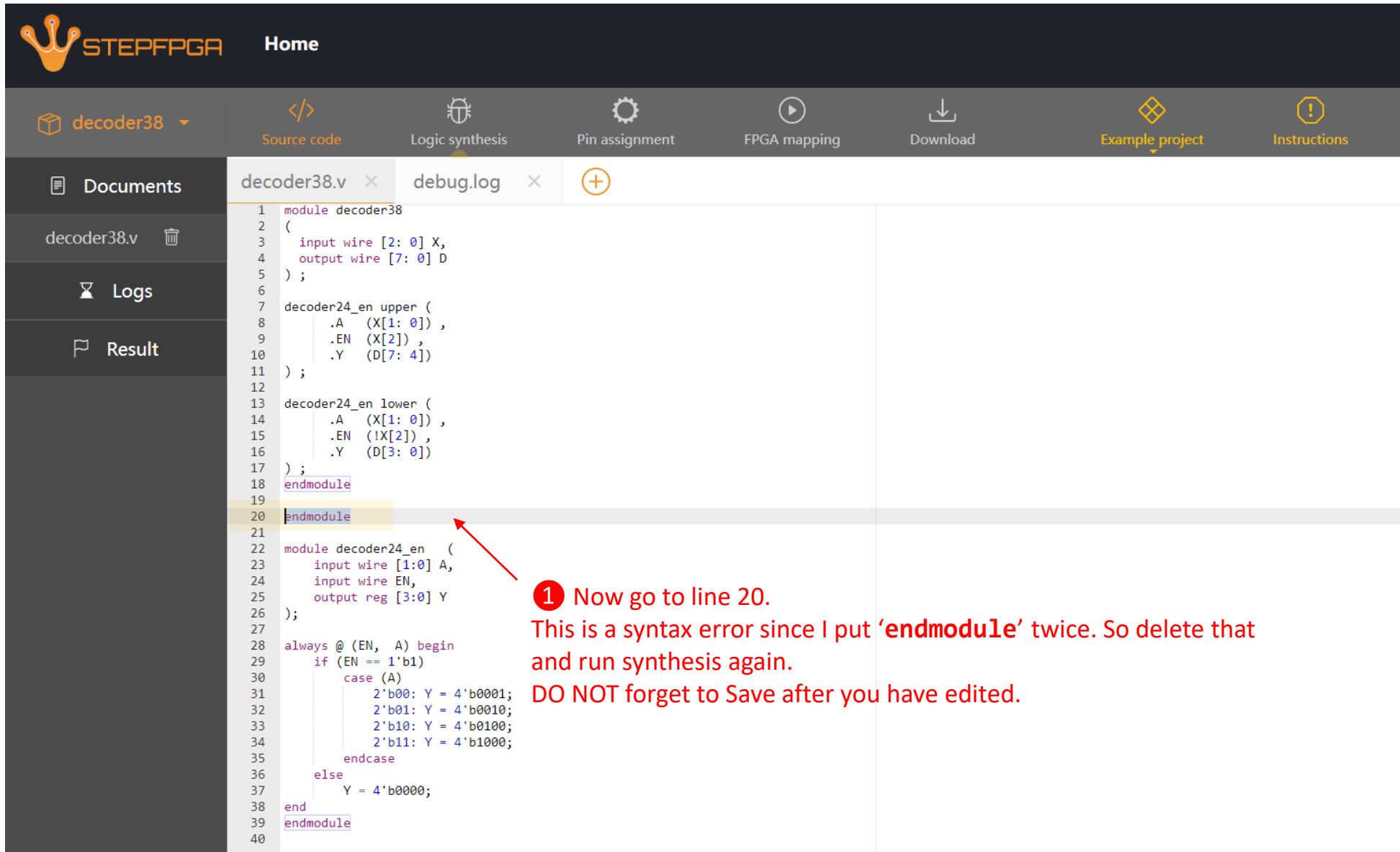
```
debug.log  
decoder38.v:20: syntax error  
I give up.
```

Two red annotations are present:

- 1** Error detected. This means you have syntax errors occurred at line 20. So you need go back to your design file
- 2** You can always click and check your design file and debug files here.



# Trace your error



The screenshot shows the STEPPFGA web interface. The top navigation bar includes the logo and the word "Home". Below it is a toolbar with icons for Source code, Logic synthesis, Pin assignment, FPGA mapping, Download, Example project, and Instructions. The main area displays a code editor with two tabs: "decoder38.v" and "debug.log". The code in "decoder38.v" is as follows:

```
1 module decoder38
2 (
3   input wire [2: 0] X,
4   output wire [7: 0] D
5 );
6
7 decoder24_en upper (
8   .A (X[1: 0]) ,
9   .EN (X[2]) ,
10  .Y (D[7: 4])
11 );
12
13 decoder24_en lower (
14   .A (X[1: 0]) ,
15   .EN (!X[2]) ,
16   .Y (D[3: 0])
17 );
18 endmodule
19
20 endmodule
21
22 module decoder24_en (
23   input wire [1:0] A,
24   input wire EN,
25   output reg [3:0] Y
26 );
27
28 always @ (EN, A) begin
29   if (EN == 1'b1)
30     case (A)
31       2'b00: Y = 4'b0001;
32       2'b01: Y = 4'b0010;
33       2'b10: Y = 4'b0100;
34       2'b11: Y = 4'b1000;
35     endcase
36   else
37     Y = 4'b0000;
38 end
39 endmodule
40
```

A red arrow points to line 20, which contains a duplicate `endmodule` statement. A red text box with a circled "1" contains the following instructions:

1 Now go to line 20.  
This is a syntax error since I put 'endmodule' twice. So delete that  
and run synthesis again.  
DO NOT forget to Save after you have edited.

# Continue to Pin Assignment

STEPFPGA Home

decoder38

Source code Logic synthesis Pin assignment FPGA mapping Download Example project Instructions

Documents decoder38.v debug.log

Logs Result

debug.log

Success

② Pin Assignment maps the inputs and outputs of the module to physical pins on FPGA board

① Success! Now we are good to move on.

# Visualize the Pin Assignment with one glimpse

STEPFFPGA Home

decoder38

Source code Logic synthesis Pin assignment FPGA mapping Download Example project Instructions

Documents

decoder38.v

Logs

Result

Pin assignment

Board

STEP-MXO2 V2.2

STEP-MXO2Core

Internal External

Key

Key	Input
KEY1	L14 X
KEY2	M13 X
KEY3	M14 X
KEY4	N14 X

Clock

Clock	Input
PCLK	C1 X

RGB\_LED

RGB_LED	Output
R_LED1	M2 X
G_LED1	N2 X
B_LED1	P2 X
R_LED2	M3 X
G_LED2	N3 X
B_LED2	P4 X

Switch

Switch	Input
X[0]	M7 X
X[1]	M8 X
X[2]	M9 X
Switch4	M10 X

LED

LED	Output
D[0]	N13 X
D[1]	M12 X
D[2]	P12 X
D[3]	M11 X
D[4]	P11 X
D[5]	N10 X
D[6]	N9 X
D[7]	P9 X

SEG1

SEG1	Output
SEG-A1	A10 X
SEG-B1	C11 X
SEG-C1	F2 X
SEG-D1	E1 X
SEG-E1	E2 X
SEG-F1	A9 X
SEG-G1	B9 X
SEG-DP1	F1 X
SEG-DIG1	C9 X

SEG2

SEG2	Output
SEG-A2	C12 X
SEG-B2	B14 X
SEG-C2	J1 X
SEG-D2	H1 X
SEG-E2	H2 X
SEG-F2	B12 X
SEG-G2	A11 X
SEG-DP2	K1 X
SEG-DIG2	A12 X

Reset Save

- 1 Make sure you select the correct Board.
  - MXO2Core has TYP-C connector
  - MXO2 V2.2 has MicroUSB connector

3 Assign all inputs and outputs of the module to FPGA pins

4 Save after done

# Check for FPGA internal routing summary information

The screenshot shows the STEP-FPGA web interface. The top navigation bar includes the STEP-FPGA logo and the word "Home". Below this is a secondary navigation bar with icons and labels for "Source code", "Logic synthesis", "Pin assignment", "FPGA mapping", "Download", "Example project", and "Instructions". The "FPGA mapping" icon is highlighted with a red arrow and a circled "1".

Below the navigation bar is a tabbed interface with tabs for "decoder38.v", "debug.log", "run.log", and a "+" icon. The "run.log" tab is active and highlighted with a red arrow and a circled "2".

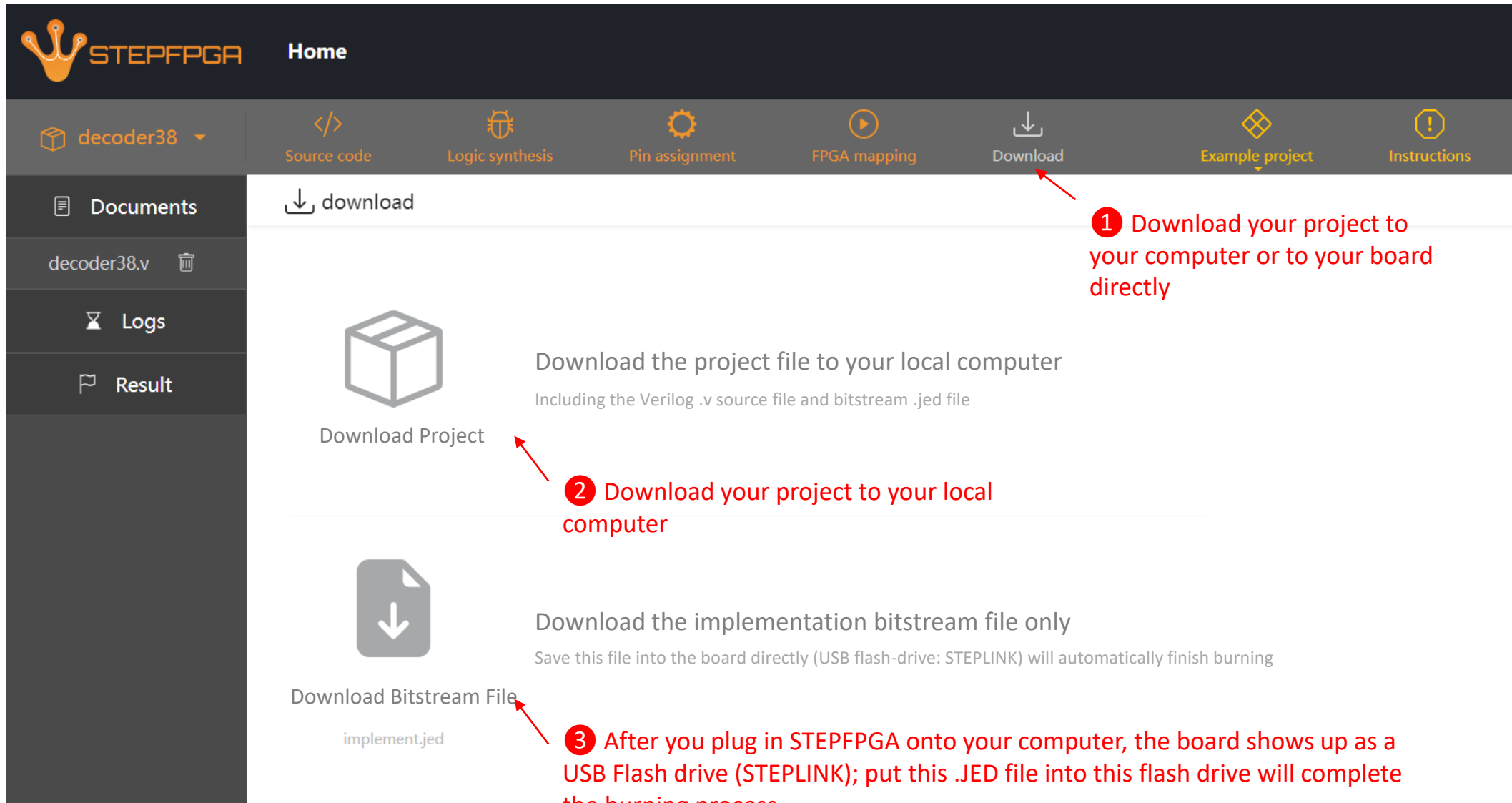
The main content area displays the contents of the "run.log" file. The text includes:

```
run.log  
synthesis -f "PRJ_1344_implement_lattice.synproj"  
synthesis: version Diamond (64-bit) 3.12.0.240.2  
  
Copyright (c) 1991-1994 by NeoCAD Inc. All rights reserved.  
Copyright (c) 1995 AT&T Corp. All rights reserved.  
Copyright (c) 1995-2001 Lucent Technologies Inc. All rights reserved.  
Copyright (c) 2001 Agere Systems All rights reserved.  
Copyright (c) 2002-2020 Lattice Semiconductor Corporation, All rights reserved.  
Mon May 16 07:17:09 2022  
  
Command Line: synthesis -f PRJ_1344_implement_lattice.synproj  
  
INFO - synthesis: Lattice Synthesis Engine Launched.  
Synthesis options:  
The -a option is MachXO2.  
The -s option is 5.  
The -t option is CSBGA132.  
The -d option is LCMXO2-4000HC.  
Using package CSBGA132.  
Using performance grade 5.  
  
#####  
  
### Lattice Family : MachXO2  
  
### Device : LCMXO2-4000HC  
  
### Package : CSBGA132
```

2 Check the run.log for debugging information and FPGA design summary

1 This step converts bitstream files that can be loaded into STEP-FPGA board for internal routing

# Burn the code into STEP FPGA board



The screenshot shows the STEP FPGA web interface. At the top left is the STEP FPGA logo and the word "Home". Below this is a navigation bar with icons for Source code, Logic synthesis, Pin assignment, FPGA mapping, Download, Example project, and Instructions. The "Download" icon is highlighted with a red arrow and a red circle containing the number 1. On the left side, there is a sidebar with "Documents" containing "decoder38.v", "Logs", and "Result". The main content area shows a "download" button with a red arrow pointing to it. Below this are three options: "Download Project" (with a red arrow and a red circle containing the number 2), "Download the implementation bitstream file only" (with a red arrow pointing to the "Download Bitstream File" link and a red circle containing the number 3), and "Download Bitstream File" (with a red arrow pointing to the "implement.jed" link).

**1** Download your project to your computer or to your board directly

**2** Download your project to your local computer

**3** After you plug in STEP FPGA onto your computer, the board shows up as a USB Flash drive (STEPLINK); put this .JED file into this flash drive will complete the burning process.