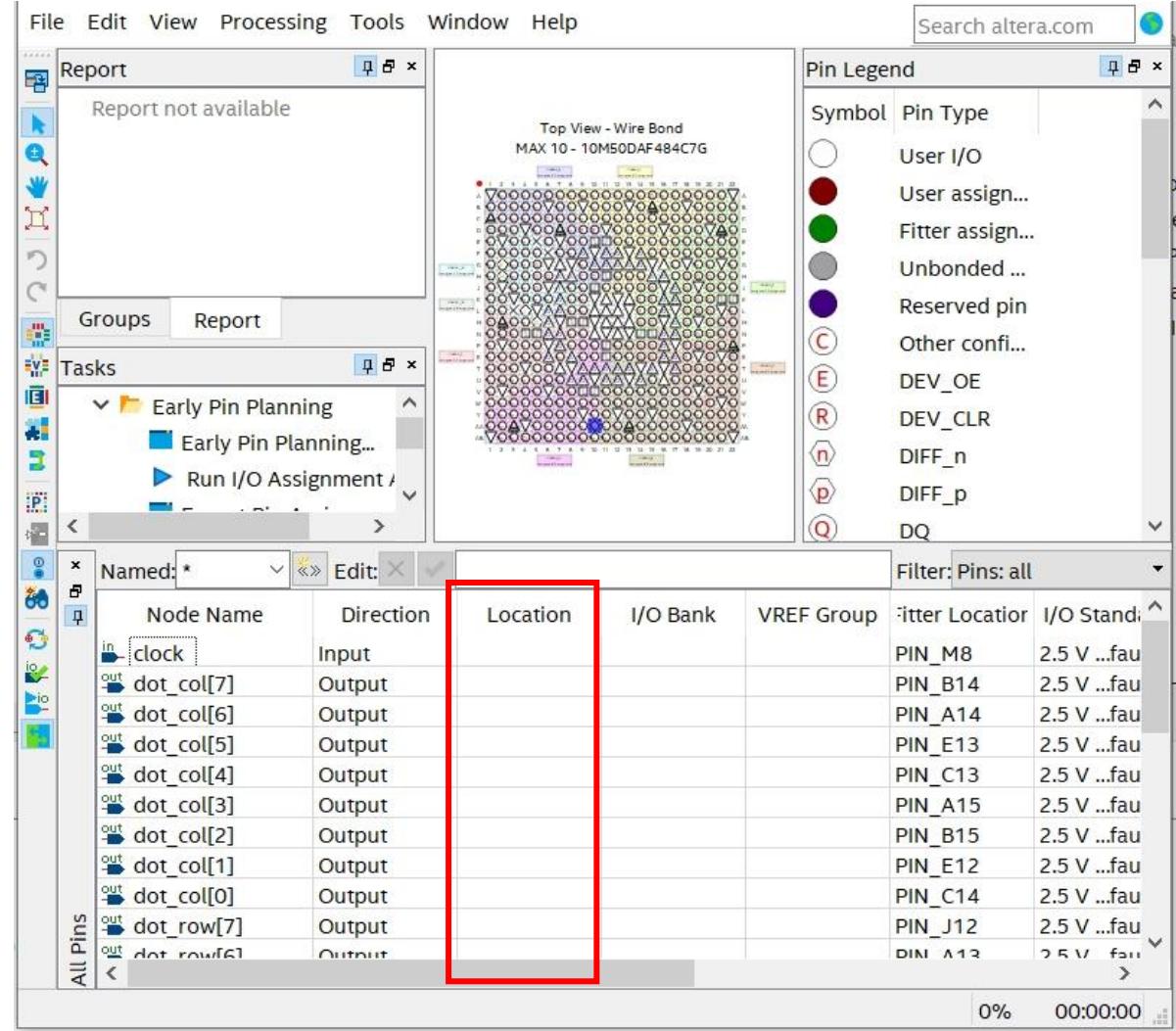


# Auto Pin Plan on quartus

DICLAB

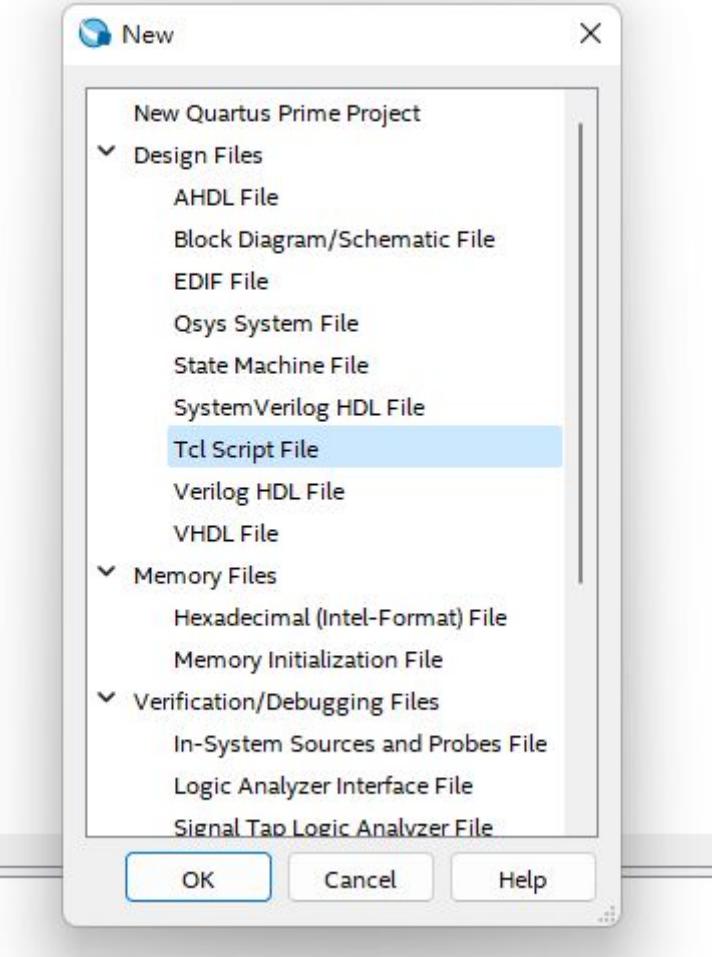


過去的lab中，會需要同學手動輸入Pin腳位

The screenshot shows the Quartus II software interface. The menu bar includes File, Edit, View, Project, Assignments, Processing, Tools, Window, and Help. The main window displays a VHDL code for a seven-segment display module. The code defines a module `sevenDisplay` with inputs `in` (3-bit) and output `out` (6-bit). It contains a `reg` variable `out` and an `always` block with a `case` statement mapping 16 input values to their corresponding 6-bit binary outputs. Below the code editor, several other VHDL files are listed in tabs: `clk_div_1Hz.v`, `clk_div_10000Hz.v`, `dot_matrix_controller.v`, `seven_display.v`, and `traffic_light_controller.v`. The bottom of the interface features a toolbar with various icons for design operations like simulation, synthesis, and analysis.

```
1 module sevenDisplay(in,out);
2
3   input [3:0] in;
4   output [6:0] out;
5
6   reg [6:0] out;
7
8   always@(in)
9     begin
10       case(in)
11         4'd0: out = 7'b1000000;
12         4'd1: out = 7'b1111001;
13         4'd2: out = 7'b0100100;
14         4'd3: out = 7'b0110000;
15         4'd4: out = 7'b0011001;
16         4'd5: out = 7'b0010010;
17         4'd6: out = 7'b0000010;
18         4'd7: out = 7'b1111000;
19         4'd8: out = 7'b0000000;
20         4'd9: out = 7'b0010000;
21         4'd10: out = 7'b0001000;
22         4'd11: out = 7'b0000011;
23         4'd12: out = 7'b1000110;
24         4'd13: out = 7'b0100001;
25         4'd14: out = 7'b0000110;
26         default: out = 7'b0001110;
27       endcase
28     end
29
30   endmodule
31
32
```

實務上, Quartus有提供 tcl file 執行 tcl command



新增 Tcl script 到 Project 中

```
1 #set clock
2 set_location_assignment PIN_P11 -to clock;
3
4 #set reset
5 set_location_assignment PIN_B8 -to reset;
6
7 #set dot matrix column
8 set_location_assignment PIN_W13 -to dot_col[7];
9 set_location_assignment PIN_W12 -to dot_col[6];
10 set_location_assignment PIN_AB12 -to dot_col[5];
11 set_location_assignment PIN_W6 -to dot_col[4];
12 set_location_assignment PIN_Y11 -to dot_col[3];
13 set_location_assignment PIN_W5 -to dot_col[2];
14 set_location_assignment PIN_AA14 -to dot_col[1];
15 set_location_assignment PIN_AB13 -to dot_col[0];
16
17 #set dot matrix row
18 set_location_assignment PIN_W7 -to dot_row[7];
19 set_location_assignment PIN_V8 -to dot_row[6];
20 set_location_assignment PIN_W9 -to dot_row[5];
21 set_location_assignment PIN_V7 -to dot_row[4];
22 set_location_assignment PIN_Y8 -to dot_row[3];
23 set_location_assignment PIN_W10 -to dot_row[2];
24 set_location_assignment PIN_AA10 -to dot_row[1];
25 set_location_assignment PIN_AA8 -to dot_row[0];
```

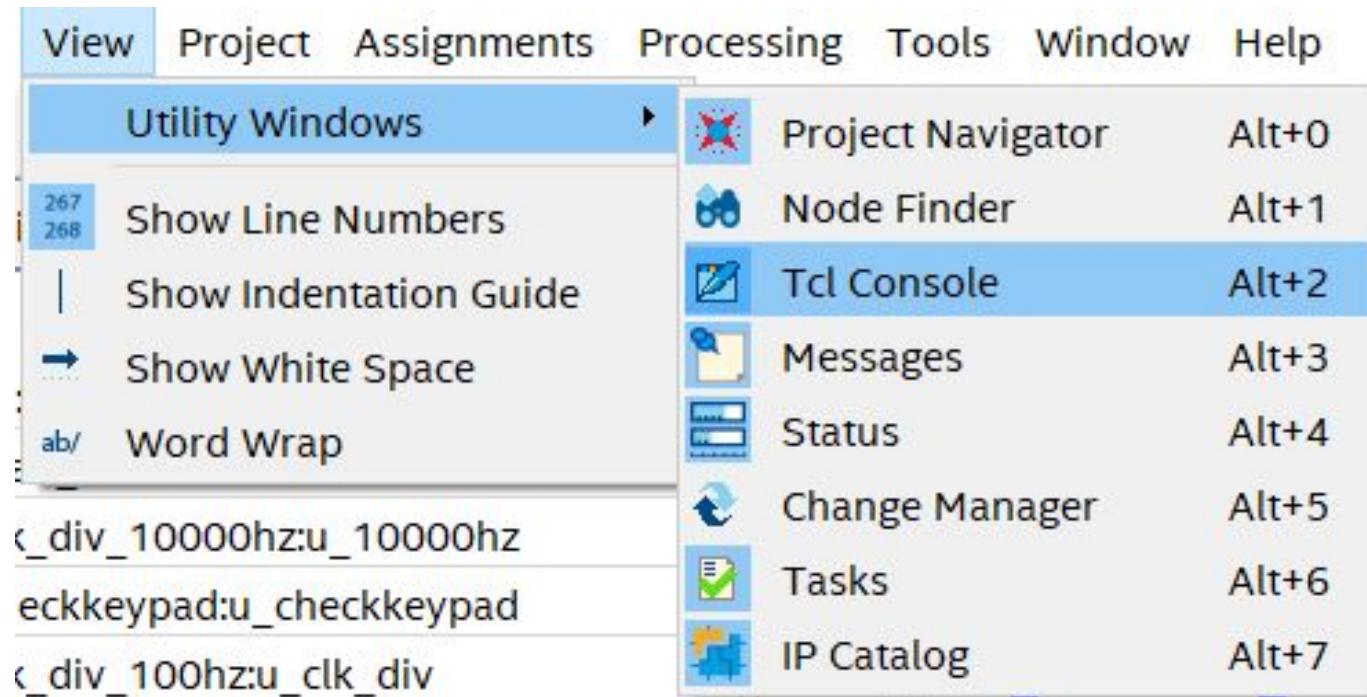
就可以開始撰寫想要的Tcl command

因此可以針對已知的Pin腳位寫上location

例如: dot matrix很多project都會用到, 寫一次、寫到一個檔案就好

```
1 #set_clock
2 set_location_assignment PIN_P11 -to clock;
3
4 #set_reset
5 set_location_assignment PIN_B8 -to reset;
6
7 #set_dot_matrix_column
8 set_location_assignment PIN_W13 -to dot_col[7];
9 set_location_assignment PIN_W12 -to dot_col[6];
10 set_location_assignment PIN_AB12 -to dot_col[5];
11 set_location_assignment PIN_W6 -to dot_col[4];
12 set_location_assignment PIN_Y11 -to dot_col[3];
13 set_location_assignment PIN_W5 -to dot_col[2];
14 set_location_assignment PIN_AA14 -to dot_col[1];
15 set_location_assignment PIN_AB13 -to dot_col[0];
16
17 #set_dot_matrix_row
18 set_location_assignment PIN_W7 -to dot_row[7];
19 set_location_assignment PIN_V8 -to dot_row[6];
20 set_location_assignment PIN_W9 -to dot_row[5];
21 set_location_assignment PIN_V7 -to dot_row[4];
22 set_location_assignment PIN_Y8 -to dot_row[3];
23 set_location_assignment PIN_W10 -to dot_row[2];
24 set_location_assignment PIN_AA10 -to dot_row[1];
25 set_location_assignment PIN_AA8 -to dot_row[0];
26
27 #set_keypad_column
28 set_location_assignment PIN_AA2 -to keypad_col[3];
29 set_location_assignment PIN_Y3 -to keypad_col[2];
30 set_location_assignment PIN_Y4 -to keypad_col[1];
31 set_location_assignment PIN_Y5 -to keypad_col[0];
32
33 #set_keypad_row
34 set_location_assignment PIN_AA6 -to keypad_row[3];
35 set_location_assignment PIN_AA5 -to keypad_row[2];
36 set_location_assignment PIN_AB3 -to keypad_row[1];
37 set_location_assignment PIN_AB2 -to keypad_row[0];
```

以KeyPad controller lab 為例的 TCL file



把pin\_plan存檔後

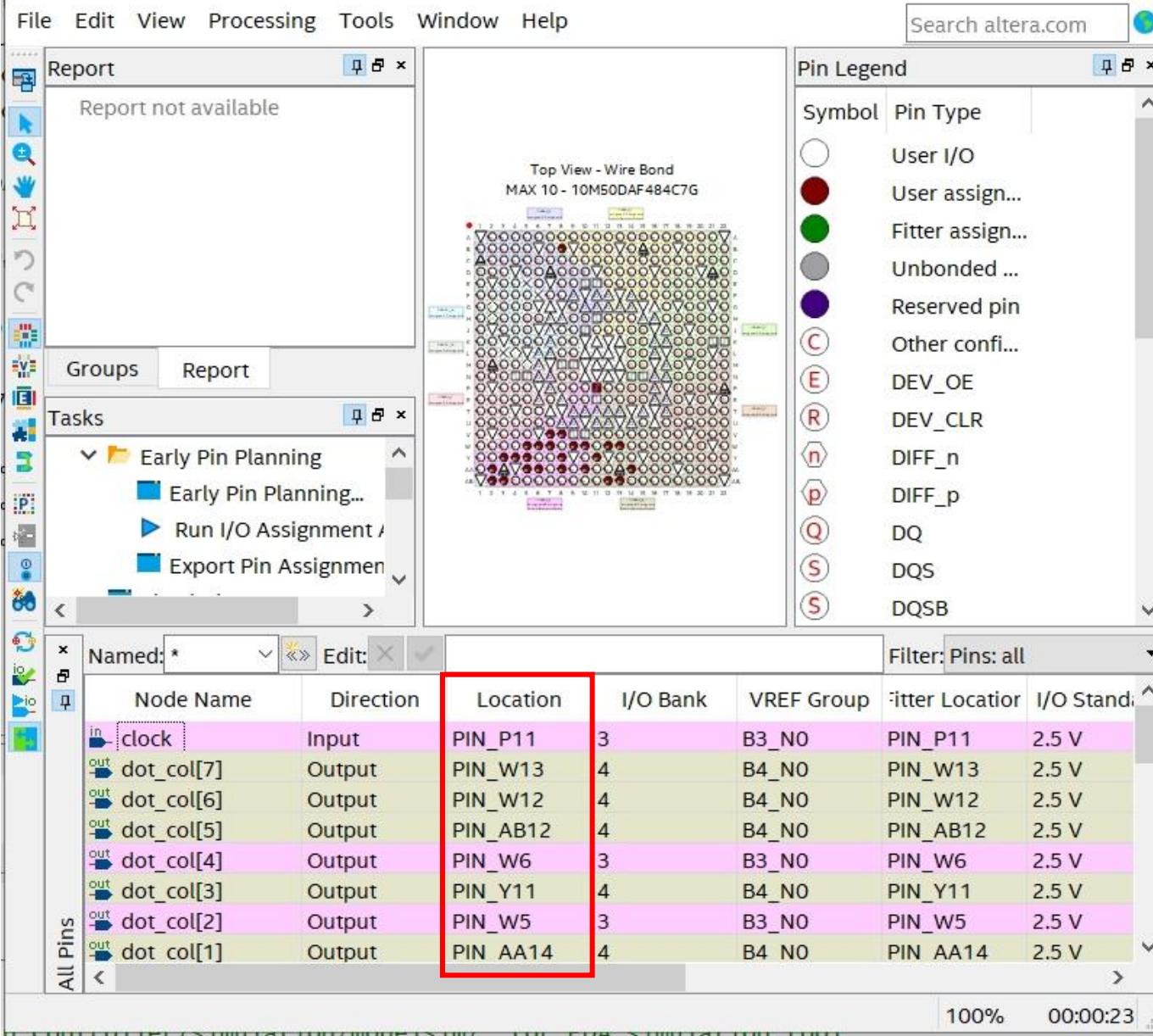
View -> Utility Windows -> Tcl Console

The screenshot shows the Quartus Prime interface. On the left, there's a 'Tasks' panel with various options like 'Compile Design', 'Analysis & Synthesis', and 'Assembler (Generate program)'. The main area displays a portion of a Tcl script:

```
23 set_location_assignment PIN_B12 -to out_row[1];
24 set_location_assignment PIN_E15 -to out_row[0];
25
26 # set seven digit output
27 set_location_assignment PIN_AA22 -to out[6];
28 set_location_assignment PIN_Y21 -to out[5];
29 set_location_assignment PIN_Y22 -to out[4];
30 set_location_assignment PIN_W21 -to out[3];
31 set_location_assignment PIN_W22 -to out[2];
32 set_location_assignment PIN_V21 -to out[1];
33 set_location_assignment PIN_U21 -to out[0];
34
35
36
37
38
39
40
41
42
43
```

A red box highlights the 'Quartus Prime Tcl Console' window at the bottom left. A red arrow points from this window down to another 'Quartus Prime Tcl Console' window below it, which contains the command `tcl> source pin_plan.tcl`.

存檔成“pin\_plan.tcl”後，使用source執行它，  
就可以節省很多時間



再開啟pin planner檢查腳位，即可完成工作