eco commands

用于在insertion添加eco逻辑进去。

不建议使用process_top_module_connections命令，这个命令是站在顶层，然后穿层次到模块里面的信号的连接，会自动开port连接，不是一个好的tile flow

1. create_instance

2. create_module

3. create_net

4. create_port

5. create_pin

6. intercept_connection

这个命令好用，用于在某个位置添加MUX或and或加buffer取反这些都很方便
而且还可以抽取ICL

```
intercept_connection node -cell_function_name  function_name
[-technology dft_cell_selection_name]
[-input2 input2_source] [-select select_source]
[-leaf_instance_prefix prefix_name]
[-only_when_has_functional_source]
```

7. create_connections

8. delete_connections

```
delete_connections obj_spec [-within] [-silent]

如果不带-within 是删除ojb instance outside外部的连线,
带-within, 就是删除ojb inside内部的连线。
```

9. move_connections

```
move_connections -from obj_spec1 -to obj_spec2 [-net_name net_name] [-
silent]

If the object within obj_spec1 a PIN:
The command moves the net connected to this pin to the pin specified in
obj_spec2 and the pin in obj_spec1 is left unconnected.
```

10. get_dft_cell

```
命令
get_dft_cell clock_buffer  -cell_selection_name tech_name_1
结果
{my_buff}
```

```
命令
get_dft_cell and -input 4
结果
{and_4}
```

eco feed through

思路：

为了ECO修改后模块尽量可重用，即模块内部修改一模一样，可以提前先放一批cell到模块里面，然后再用ECO进行连线。

connect.tar.gz

```
# set_system_mode setup
set_system_mode analysis
set_system_mode insertion

create_instance u_add_buf0 -of_module [get_dft_cell buffer] -below_instance
chnl_l
create_instance u_add_buf1 -of_module [get_dft_cell buffer] -below_instance
chnl_l

delete_connections dev0/c -silent
delete_connections dev1/c -silent
create_connections ctl/c0 chnl_l/u_add_buf0/A -net_name ft_i_0
create_connections chnl_l/u_add_buf0/Y dev0/c -net_name ft_o_0
create_connections ctl/c1 chnl_l/u_add_buf1/A -net_name ft_i_1
create_connections chnl_l/u_add_buf1/Y dev1/c -net_name ft_o_1
```

```
delete_connections dev2/c -silent
delete_connections dev3/c -silent
create_connections ctl/c2 chnl_r/u_add_buf0/A -net_name ft_i_2
create_connections chnl_r/u_add_buf0/Y dev2/c -net_name ft_o_2
create_connections ctl/c3 chnl_r/u_add_buf1/A -net_name ft_i_3
create_connections chnl_r/u_add_buf1/Y dev3/c -net_name ft_o_3

write_design -output_directory out -modified -replace
```

输出：

```
module chnl_a(ft_i_0, ft_o_0, ft_i_1, ft_o_1);
  input ft_i_0, ft_i_1;
  wire ft_i_0, ft_i_1;
  output ft_o_0, ft_o_1;
  wire ft_o_0, ft_o_1;

  buf02 u_add_buf0(
      .A(ft_i_0), .Y(ft_o_0)
  );

  buf02 u_add_buf1(
      .A(ft_i_1), .Y(ft_o_1)
  );
endmodule

  wire ft_o_0, ft_o_1, c2, c3, ft_o_2, ft_o_3;
ctl ctl(
    .c0                             (c0
), // output
    .c1                             (c1
), // output
    .c2                             (c2
), // output
    .c3                             (c3
)  // output
);

dev dev0(.c(ft_o_0));
dev dev1(.c(ft_o_1));
dev dev2(.c(ft_o_2));
dev dev3(.c(ft_o_3));

chnl_a chnl_l(.ft_i_0(c0), .ft_o_0(ft_o_0), .ft_i_1(c1), .ft_o_1(ft_o_1));
chnl_a chnl_r(.ft_i_0(c2), .ft_o_0(ft_o_2), .ft_i_1(c3), .ft_o_1(ft_o_3));
```

get nets

```
get_nets [name_patterns] [-below_instances instance_objects]
[-of_pins pin_objects | -of_ports port_objects | -of_gate_pins
gate_pin_objects |-of_pseudo_ports pseudo_port_objects] [-bundle]
[-hierarchical] [-filter attribute_equation] [-regexp] [-nocase] [-silent]


get_nets [name_patterns]  获得net名为pattern的net
get_nets -of_pins xxxpin    获取net跟pin相对应的net net名与pin名名字一样

如果想获取pin 外面连线名:
 如果是input pin可用get_fanin -stop_on net
 如果是output pin可用get_fanout -stop_on net
```