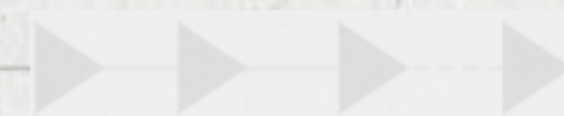


芯动力——硬件加速设计方法

第四章 逻辑综合(5)

邸志雄@西南交通大学

zxdi@home.swjtu.edu.cn





TCL的应用:1-Overview

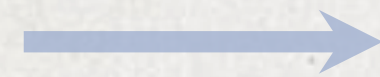
Tcl = Tool Command Language

- A simple programming language.
- Open, industry standard.
- Many Synopsys tools' command interfaces are based on the Tcl command language.



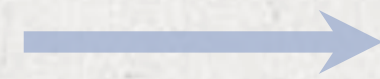
Tcl Scripts are Based on Commands

Builtin Commands



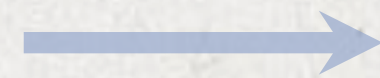
Provided by the Tcl interpreter itself

Application Commands

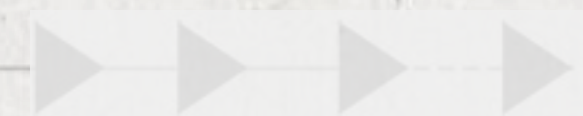


Provided by Synopsys tools, written as a command procedure in C or C++ using the Tcl extension mechanism

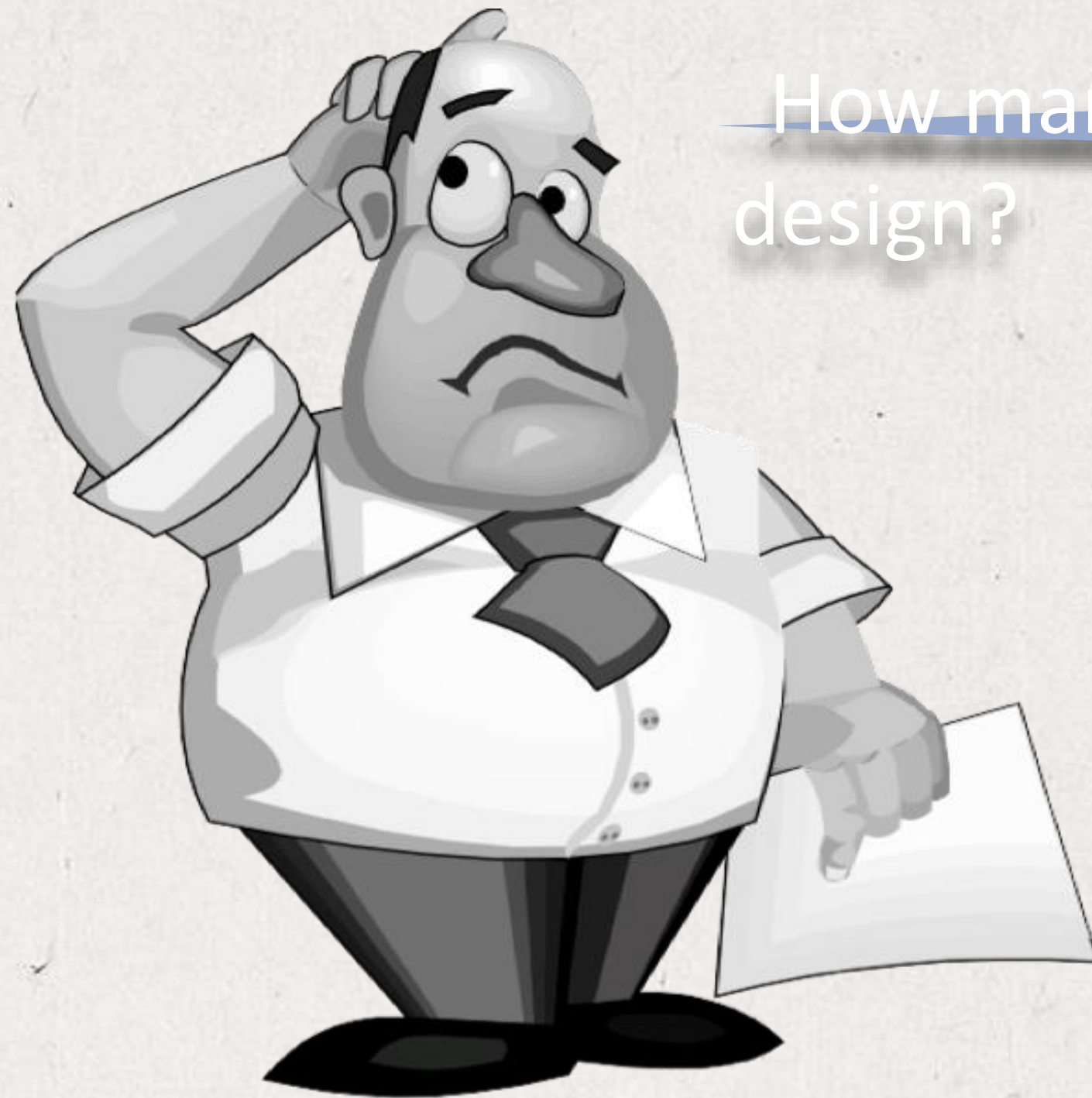
User Defined Commands



Provided by user, written as a command procedure in Tcl

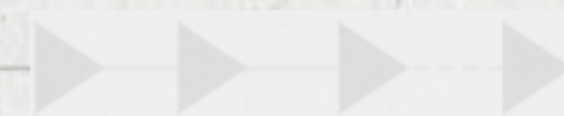


Example : Why Synopsys TCL?



How many clocks have been defined for this design?

Suggest a report that will list all clocks?



Example : Why Synopsys TCL?

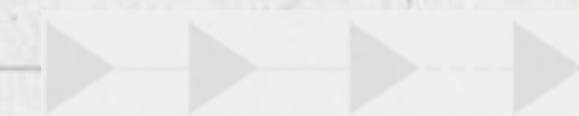
```
pt_shell> report_clock
```

Clock	Period	Waveform	Attrs	Sources
IO_PCI_CLK	15.00	{0 7.5}	p, G	{I_CLOCK_GEN/fb_clk}
IO_SDRAM_CLK	7.50	{0 3.75}	p, G	{I_CLOCK_GEN/fb_clk}
PCI_CLK	15.00	{0 7.5}	p, G	{I_CLOCK_GEN/clk}
SDRAM_CLK	7.50	{0 3.75}	p, G	{I_CLOCK_GEN/clk}
SD_DDR_CLK	7.50	{0 3.75}	p, G	{sd_CK}
SD_DDR_CLKn	7.50	{3.75 7.5}	p, G	{sd_CKn}
SYS_2x_CLK	5.00	{0 2.5}	p, G	{I_CLOCK_GEN/clk_2x}
SYS_CLK	10.00	{0 5}	p, G	{I_CLOCK_GEN/clk_1x}

...



Ridiculous! Writing
a Perl script would
be simpler!



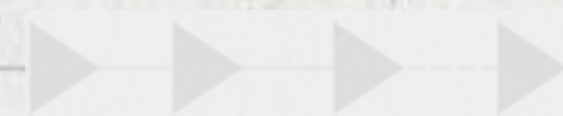
Example : Why Synopsys TCL?

```
sizeof_collection [all_clocks]  
€ 52
```

Powerful

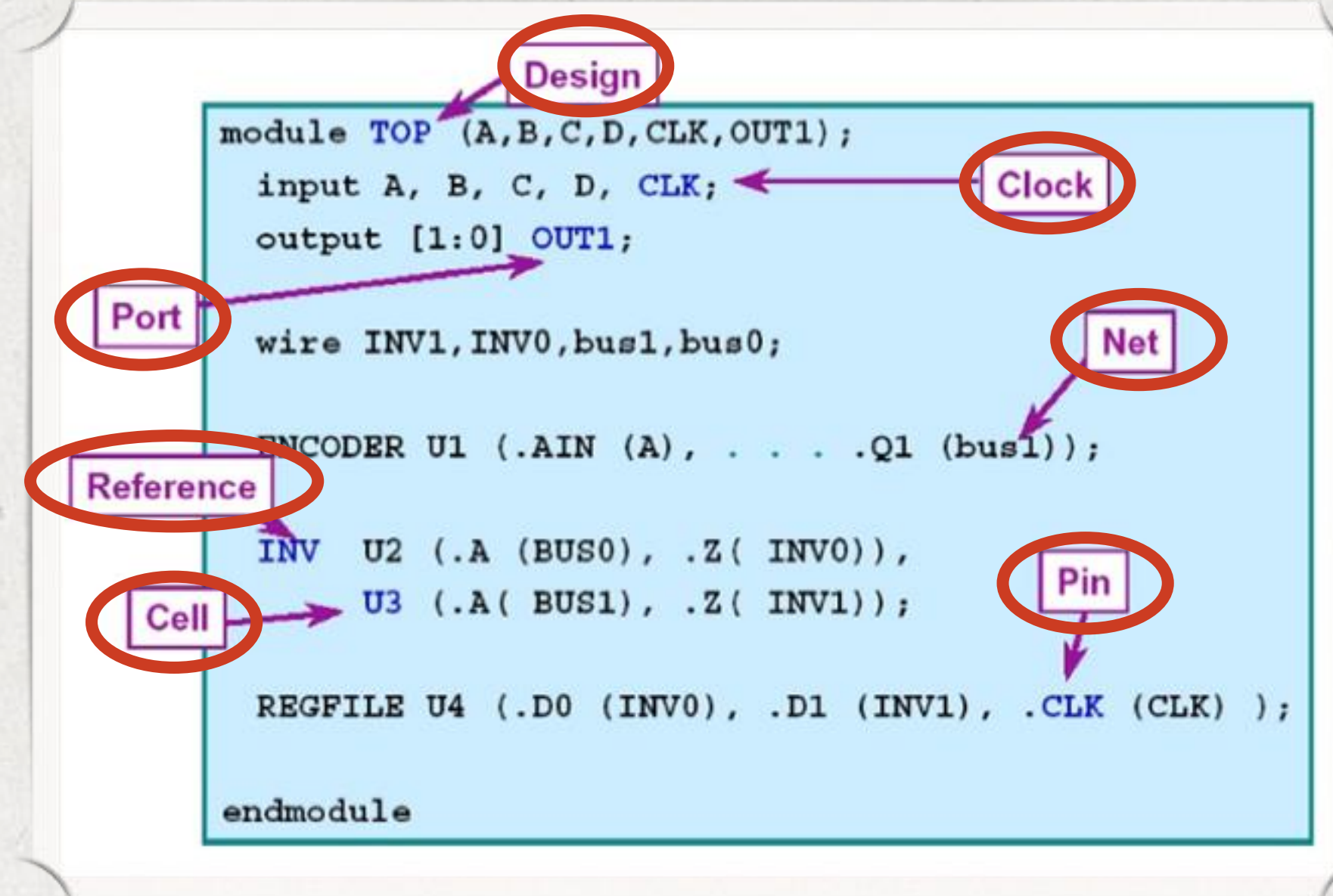
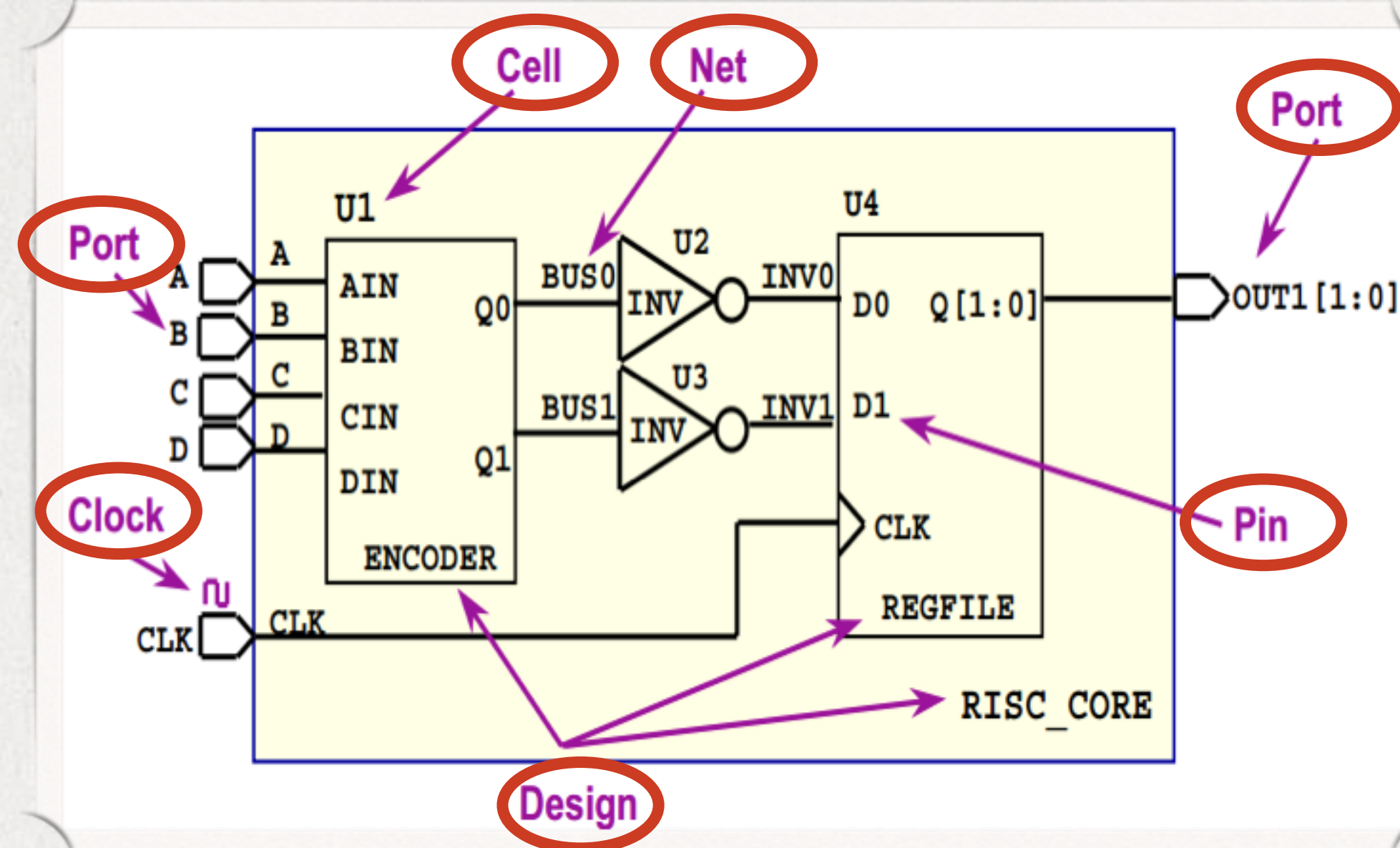
Flexible

Easy



TCL的应用: Design Object

- 通过对Design Object的分析, 我们来了解和学习DC获取电路并进行解析的方法。



Schematic View and Code of Design Objects

综合软件当中TCL的常见指令

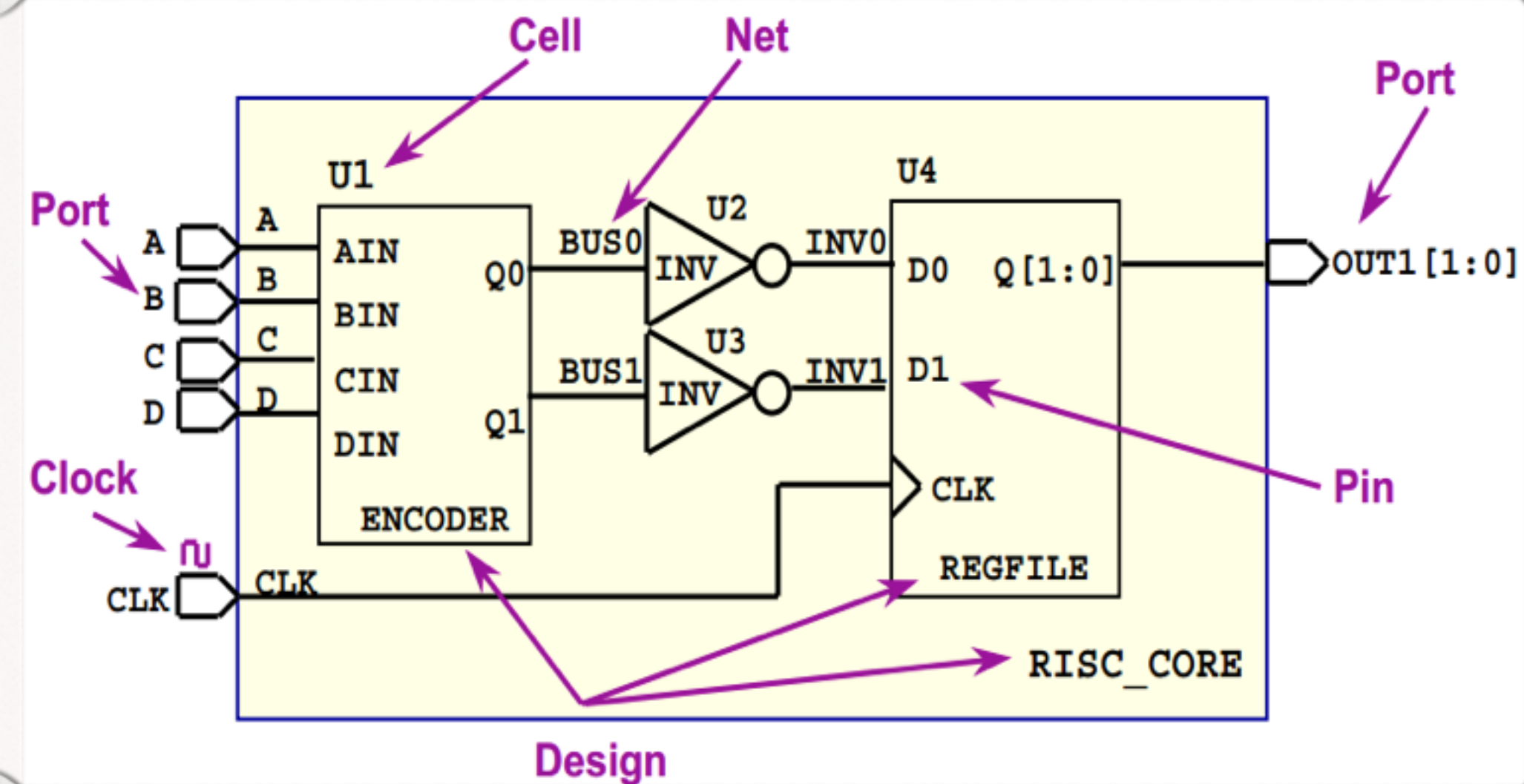
语法格式: `get_ports portsName`

指令功能: 返回design中对应的ports object

例-1: 如何查看design当中有没有一个port叫做CLK?

```
Shell> get_ports CLK
```

```
{CLK}
```



例-2：我们想查看design当中有没有一个port叫做SPI?

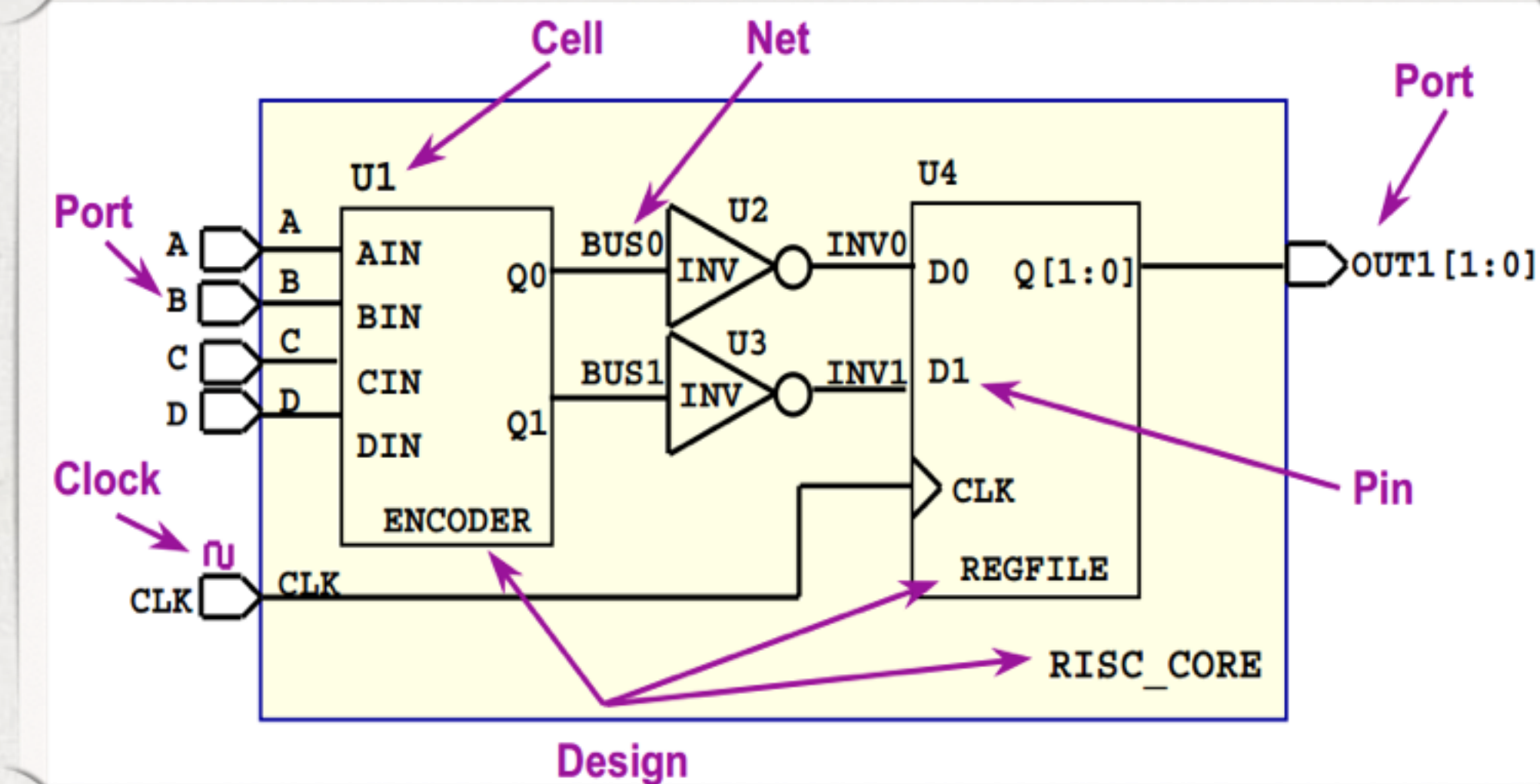
```
Shell> get_ports SPI
```

No object Found!

例-3：我们想查看design当中所有的port (*可以通配任何字符)

```
Shell> get_ports *
```

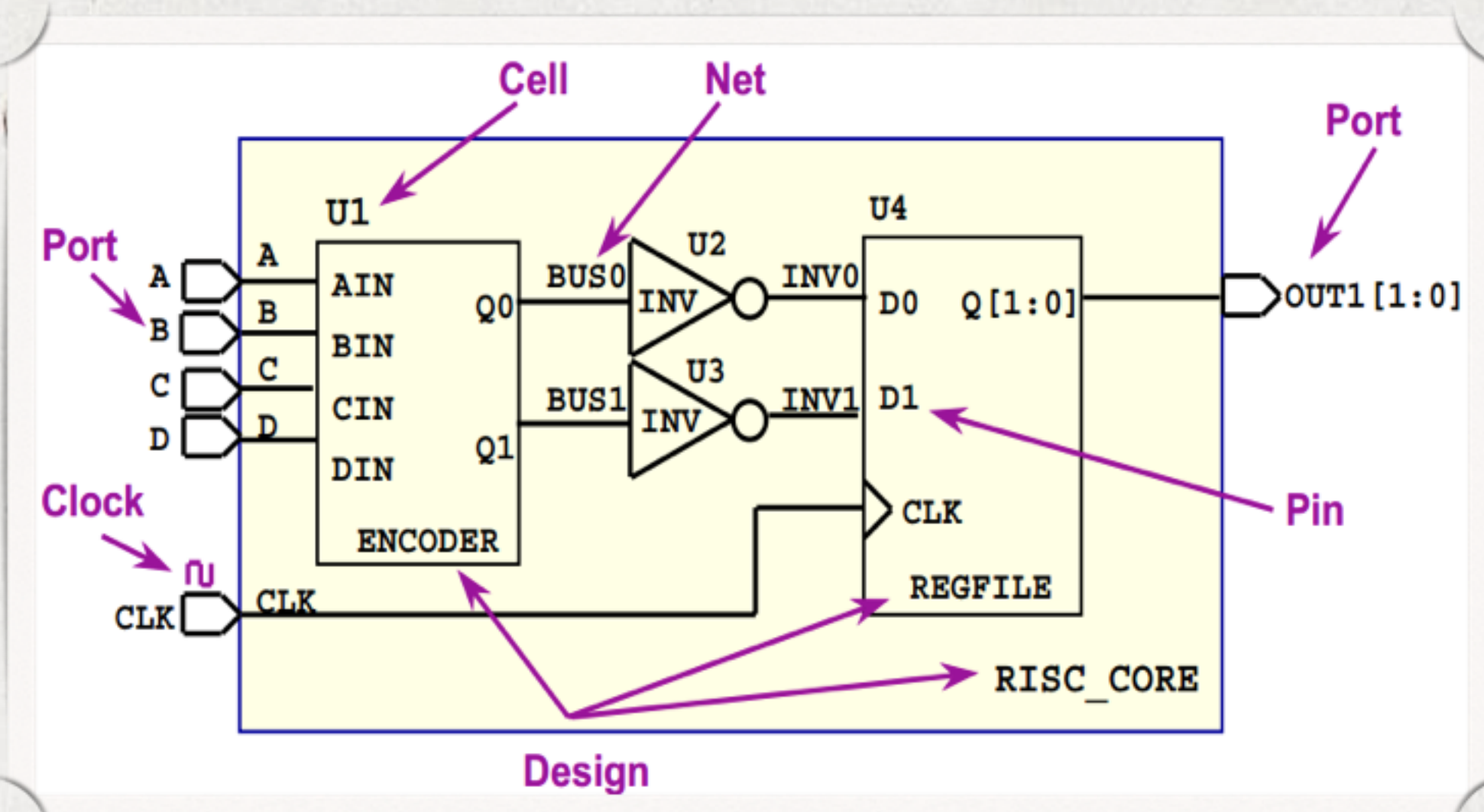
```
{A B C D CLK OUT[0] OUT[1]}
```



例-4：假设我们有port名字叫 {CLKA CLKB OUTA OUTB INA INB}
如果我们想得到所有C开头的port 怎么做？

```
Shell> get_ports C*
```

```
{CLKA CLKB}
```



综合软件当中TCL的常见指令

语法格式: `get_cells cellsName`

指令功能: 返回design中对应的cell的instance name object

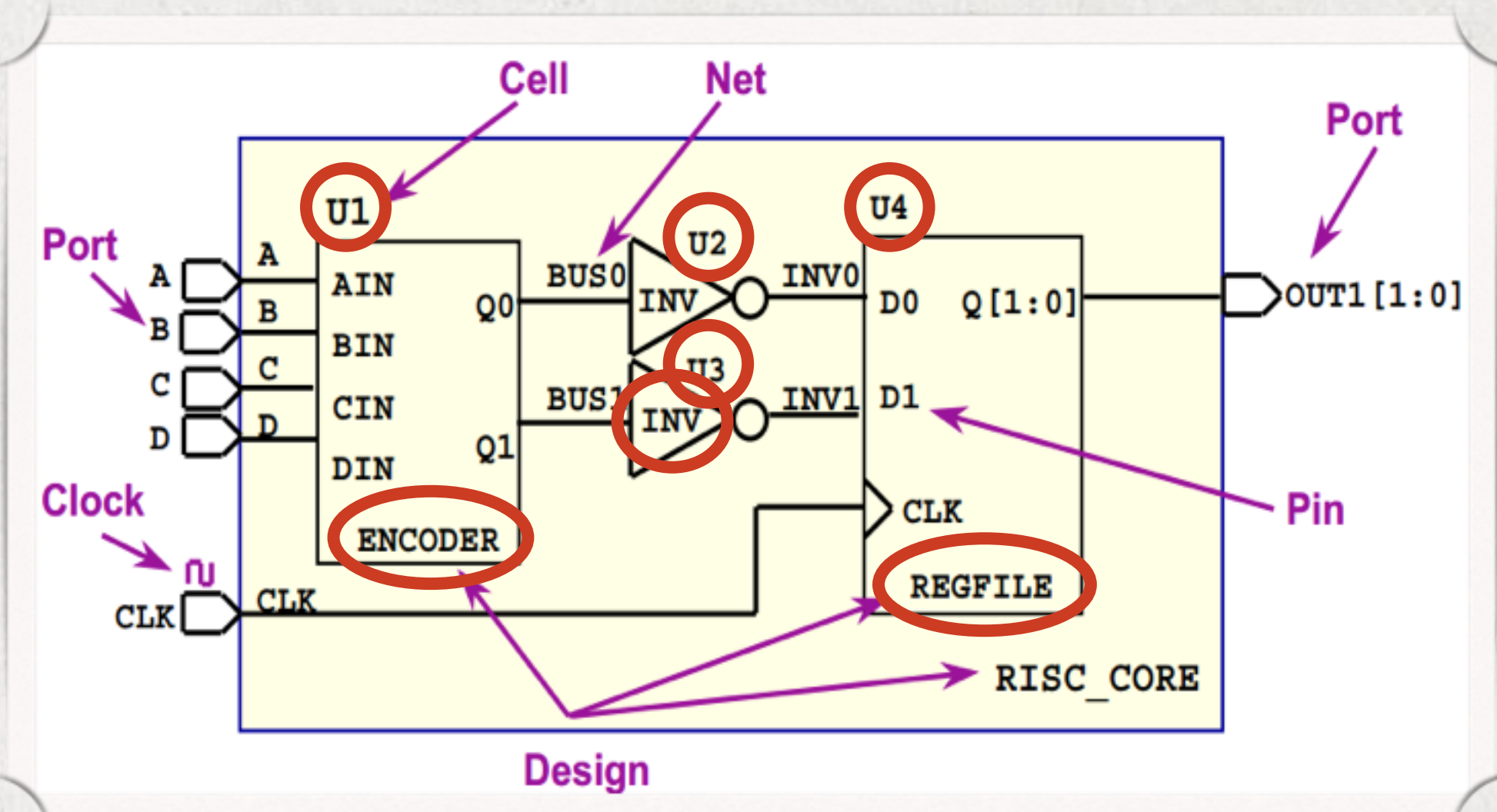
回顾几个概念:

- 什么是reference name(ref_name)?

{ENCODER INV REGFILE}

- 什么是instance name?

{U1 U2 U3 U4}



举例-1： 我们想查看design当中有没有一个cell叫做U4?

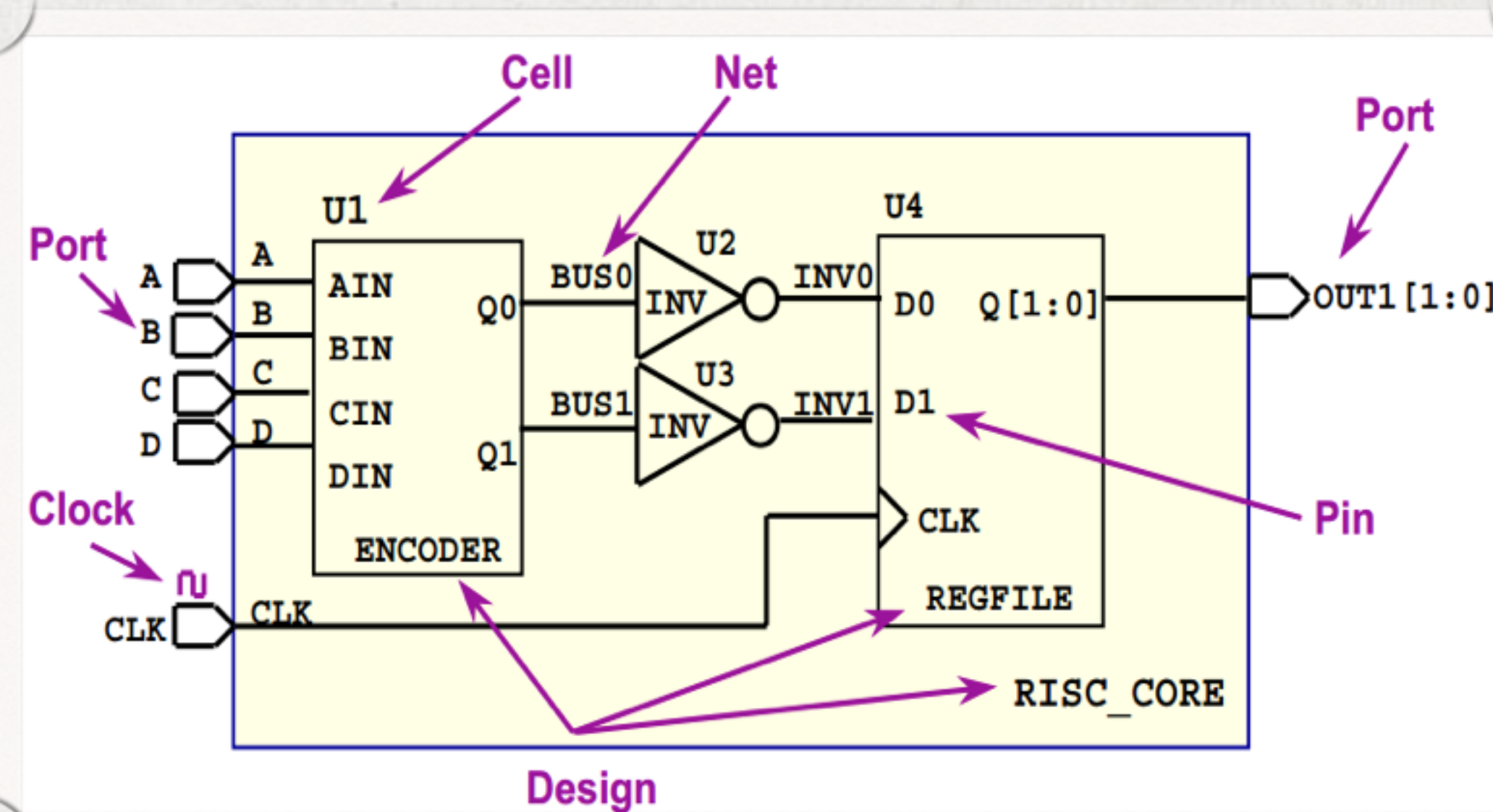
```
Shell> get_cells U4
```

```
{U4}
```

举例-2： 我们想查看design当中所有的cell

```
Shell> get_cells *
```

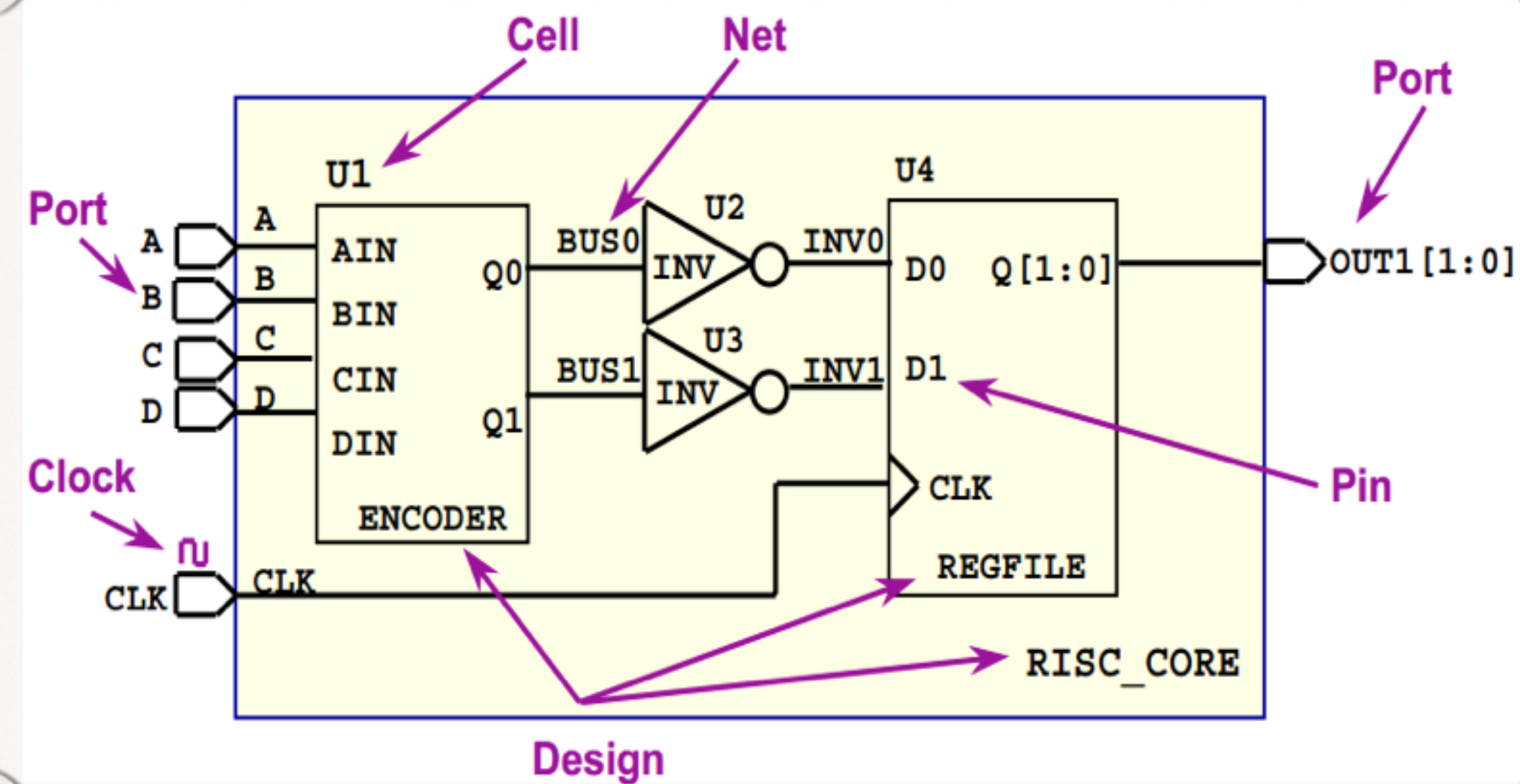
```
{U1 U2 U3 U4}
```



举例-3： 我们想查看design当中以3为结尾的cells

```
Shell> get_cells *3
```

```
{U3}
```



综合软件当中TCL的常见指令

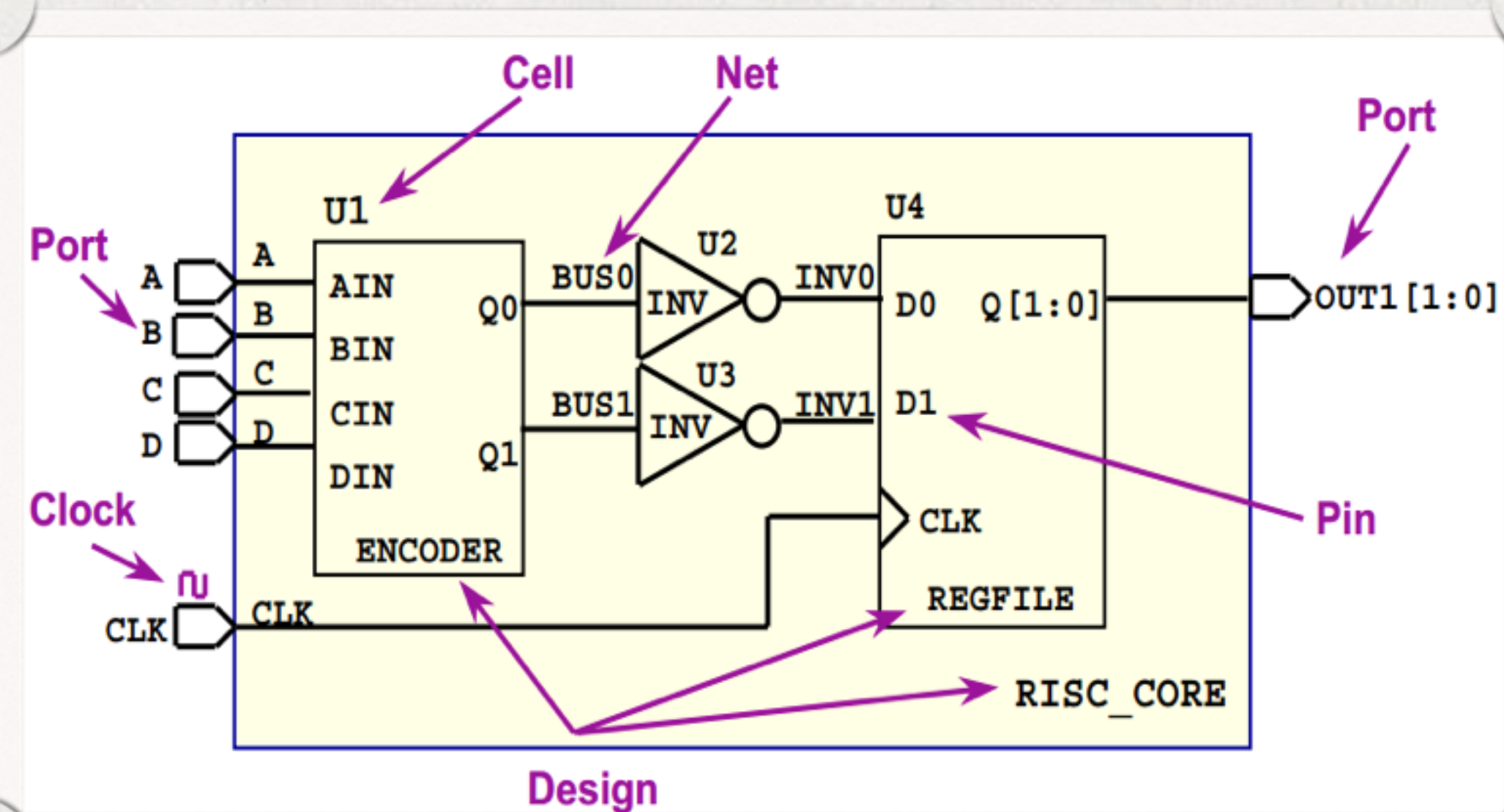
语法格式: `get_nets netsName`

指令功能: 返回design中net的object

举例-1: 查看design当中有没有一个net以INV开头?

```
Shell> get_nets INV*
```

```
{INV0 INV1}
```



举例-2: 我们想查看design当中所有的nets

```
Shell> get_nets *
```

```
{A B C D CLK BUS0 BUS1 INV0 INV1 OUT[0] OUT[1]}
```

例-3: 我们想查看design当中有多少个net?

```
Shell> llength [get_object_name get_nets *]
```

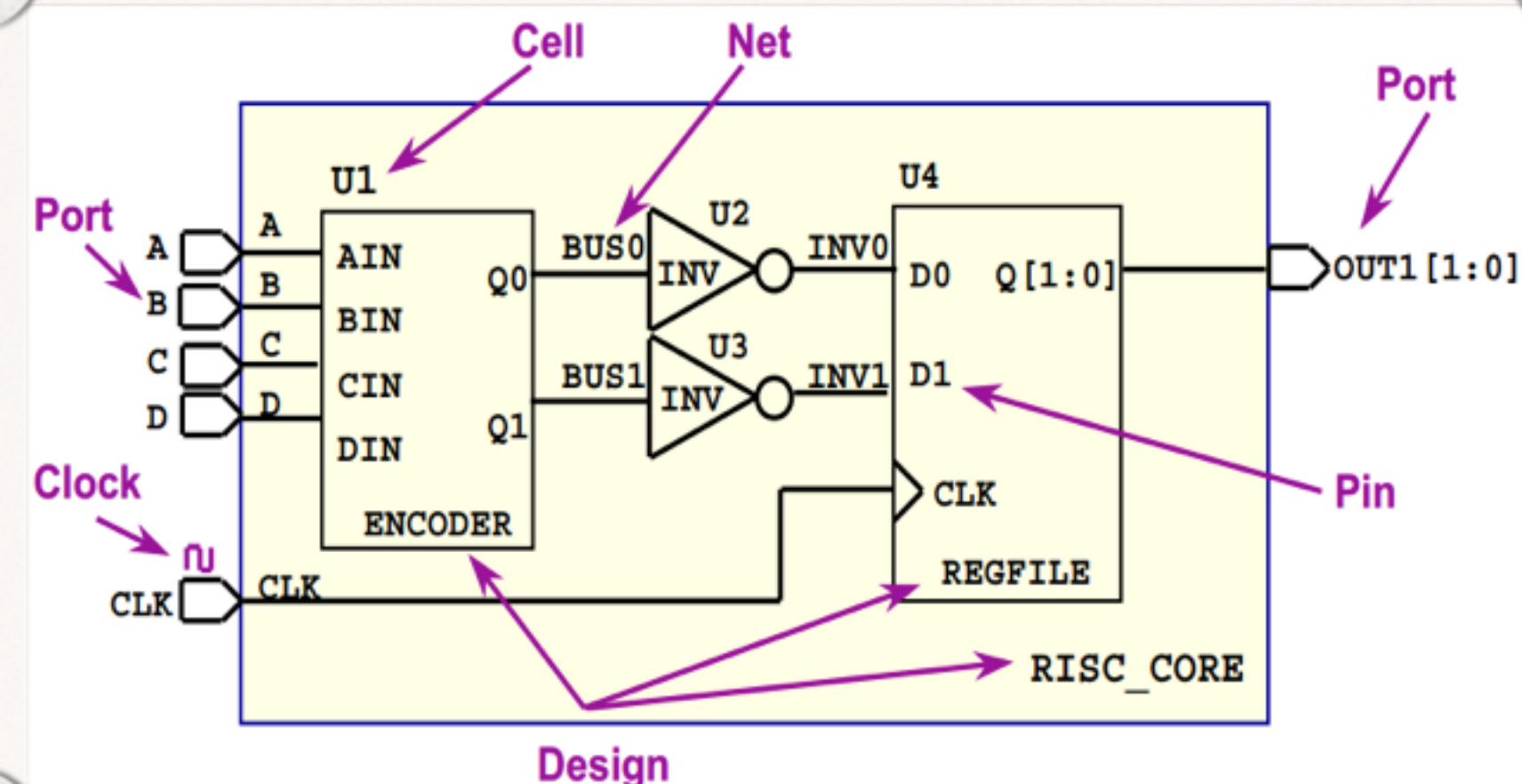
11

计算list TCL 自己本身的net的collection

```
Shell> sizeof_collection [get_nets *]
```

11

synopsys TCL



综合软件当中TCL的常见指令

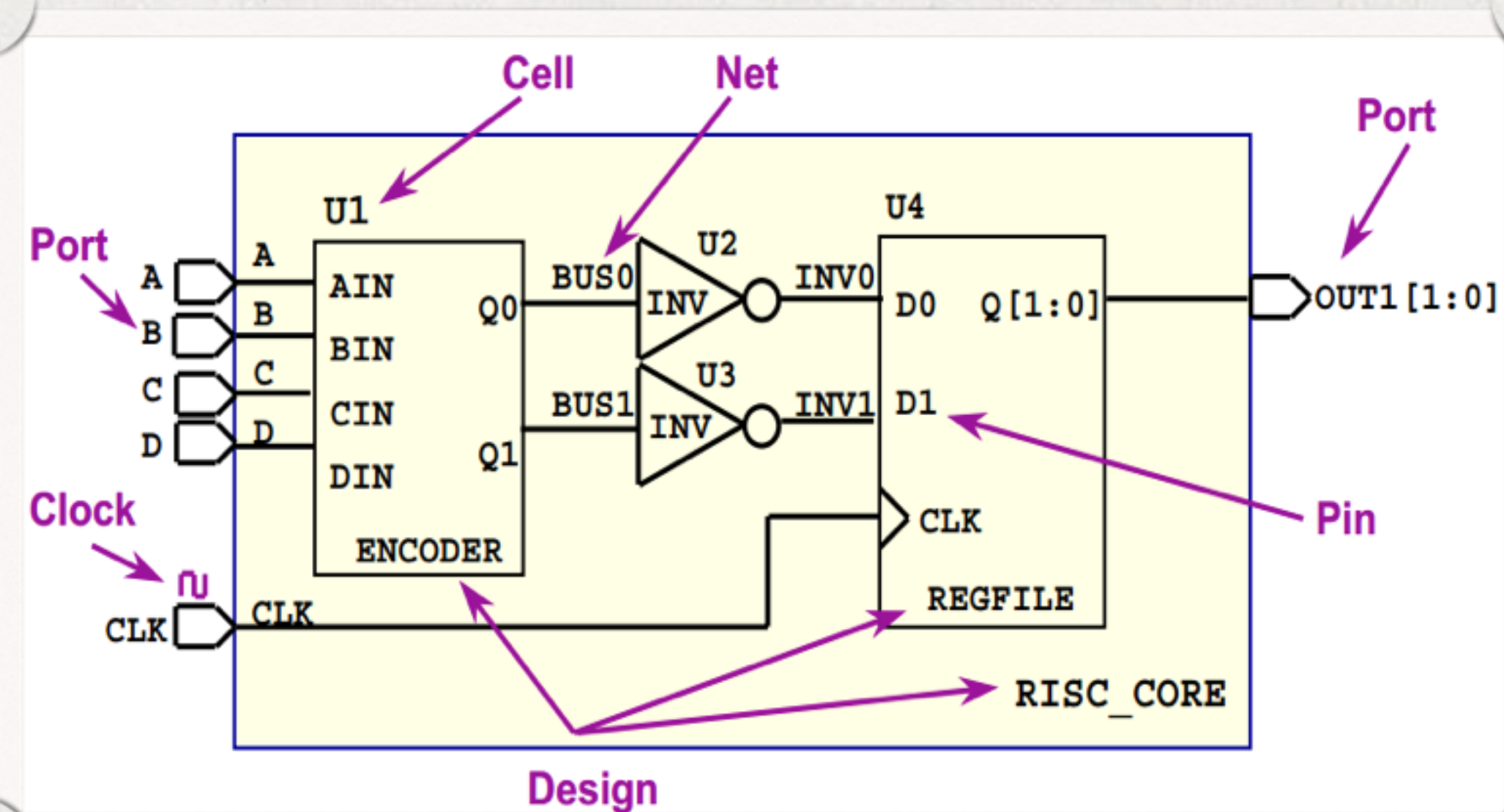
语法格式: `get_pins pinsName`

指令功能: 返回design中pin的object

举例-1: 查看design当中有哪些pin的名字叫做Z?

```
Shell> get_pins */Z
```

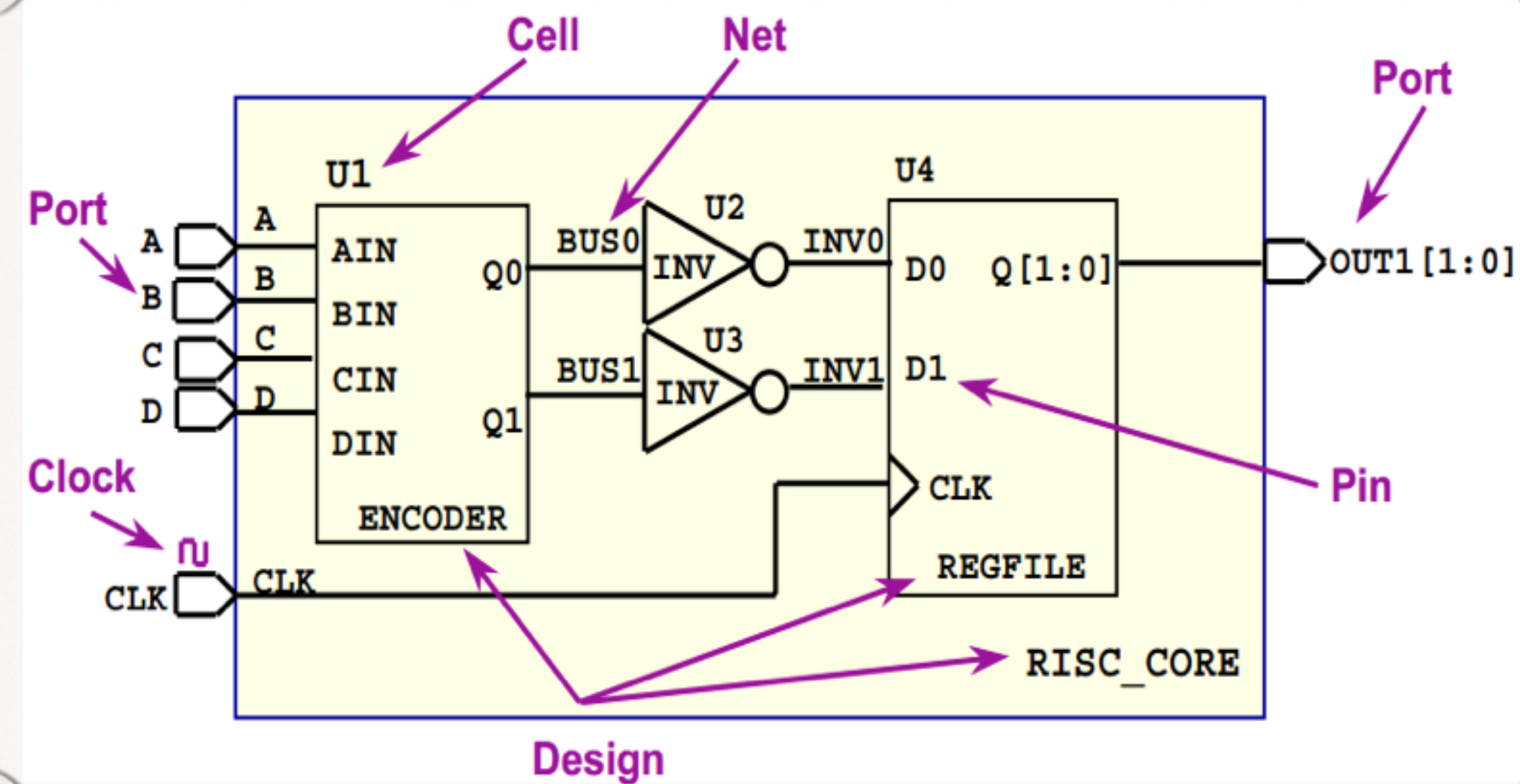
```
{INV0/Z INV1/Z}
```



举例-2: 查看design当中有哪些pin的名字以Q开头?

```
Shell> get_pins */Q*
```

```
{ENCODER/Q0 ENCODER/Q1 REGFILE/Q[1] REGFILE/Q[0]}
```



综合软件当中TCL的常见指令

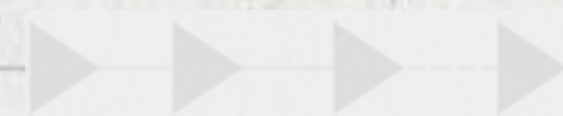
“数据类型：object（对象）” 与其 “属性”

说明：

- object对于tcl脚本一个重要的扩展
- 常见的对象有四种 cell, net, port, pin;
- 每种object有它的属性。

下面的将介绍一些常见属性：

- 任何一个属性都可以用get_attribute得到,
- list_attribute -class * 可以得到所有object 的属性,
- 部分属性可以用set_attribute来设置。

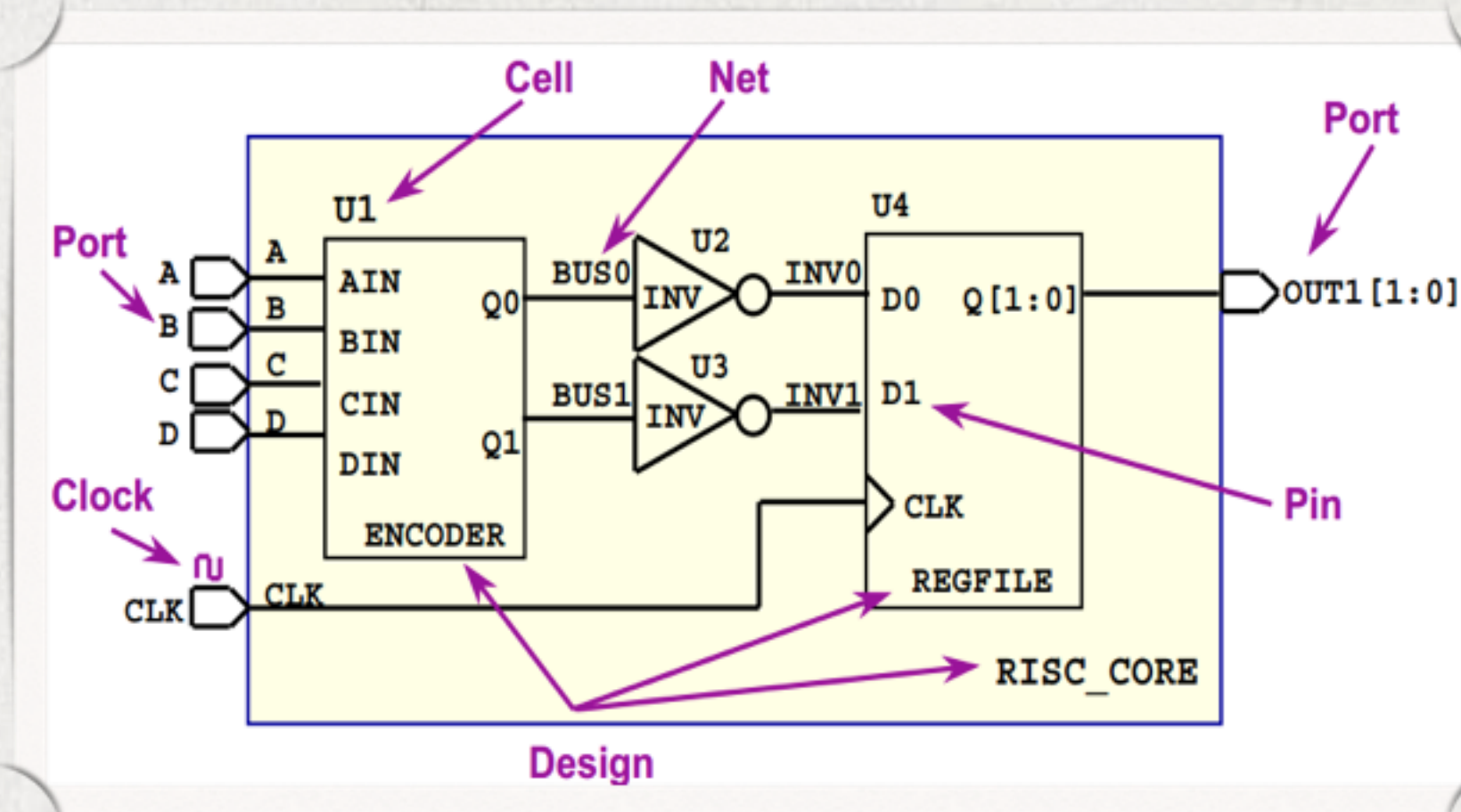


Cell object:

- 属性 `ref_name` : 用来保存其map到的reference cell名称

```
Shell> get_attribute [get_cells -h U3] ref_name
```

```
{INV}
```



Pin object:

- 属性 `owner_net` : 用来保存与之相连的net的名称

```
Shell> get_attribute [get_pins U2/A] owner_net
```

```
{BUS0}
```


Port object:

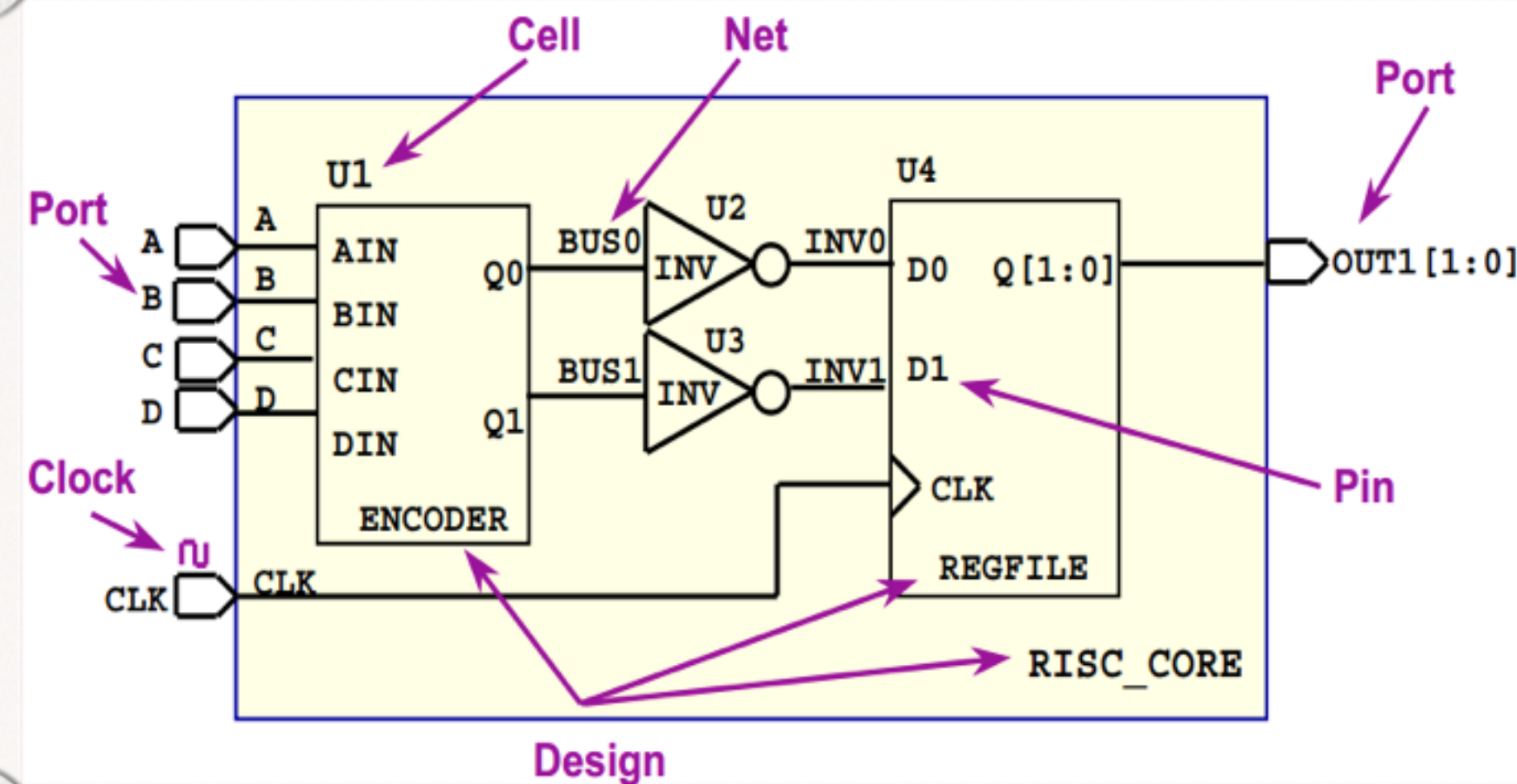
- 属性 direction：用来保存port 的方向

```
Shell> get_attribute [get_ports A] direction
```

```
{in}
```

```
Shell> get_attribute [get_ports OUT[1]] direction
```

```
{out}
```



Net object:

- 属性 full_name : 用来保存net的名称

```
Shell> get_attribute [get_nets INV0] full_name
```

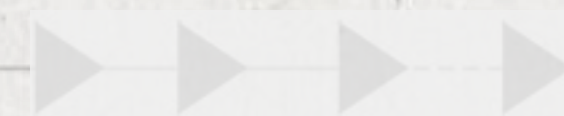
```
{INV0}
```

```
Shell> get_object_name [get_nets INV0]
```

```
{INV0}
```

```
*Shell> get_attribute INV0 full_name
```

```
Error: No attribute found
```

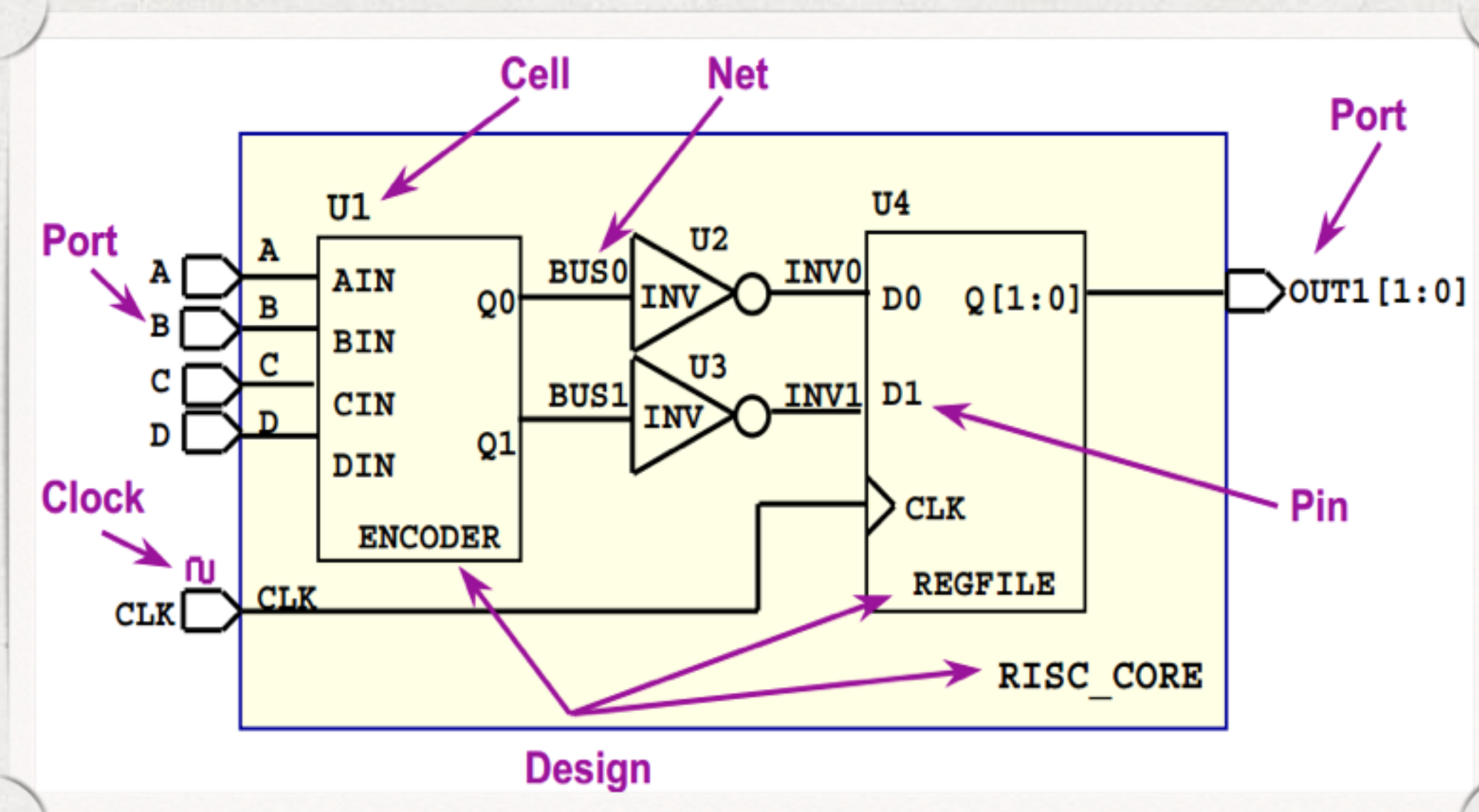


综合软件当中TCL的常见指令

理解了属性，就能做更多的事情啦：

get_* -f:

- -f 这个option可以用来过滤属性，以得到我们想要的object



例子-1：想得到所有方向是input的port

```
Shell> get_ports * -f "direction==in"
```

```
{A B C D CLK}
```

例子-2：想得到所有方向是output的pin

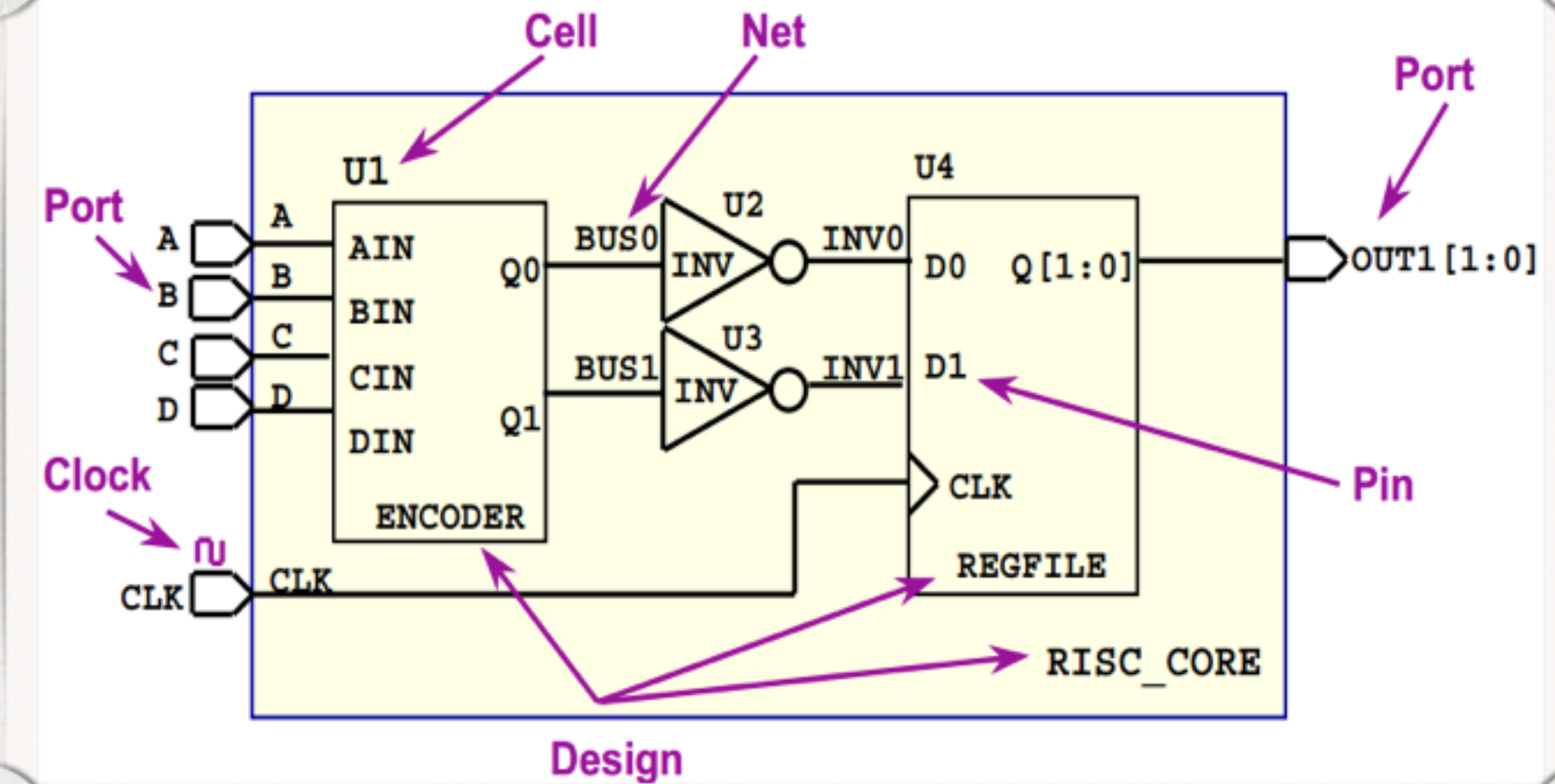
```
Shell> get_pins * -f "direction ==out"
```

```
{U1/Q0 U1/Q1 U2/Z U3/Z REGFILE/Q[0] REGFILE/Q[1]}
```

例子-3：想得到所有ref_name 是INV的 cell

```
Shell> get_cells * -f "ref_name == INV"
```

```
{U2 U3}
```

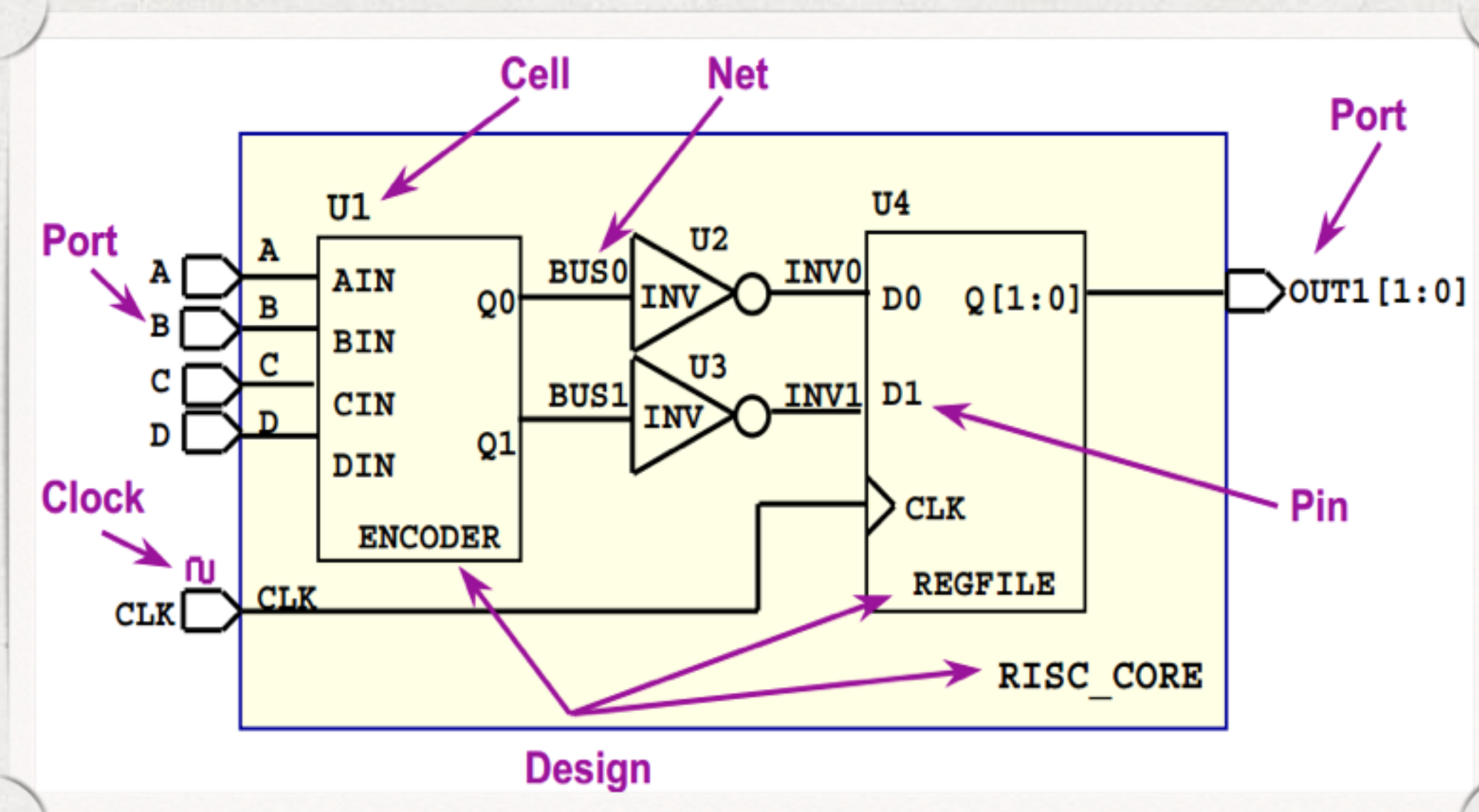


综合软件当中TCL的常见指令

另外一个最关键的option

get_* [object]-of:

- -of 这个option可以用来得到与你指定object相连接的objec



object的连接关系:

```
--port object <-> net object
```

```
>get_nets -of [get_port A]
```

A

```
> get_net -of [get_pin U2/A]
```

BUS0

```
--pin object    <->  net object
```

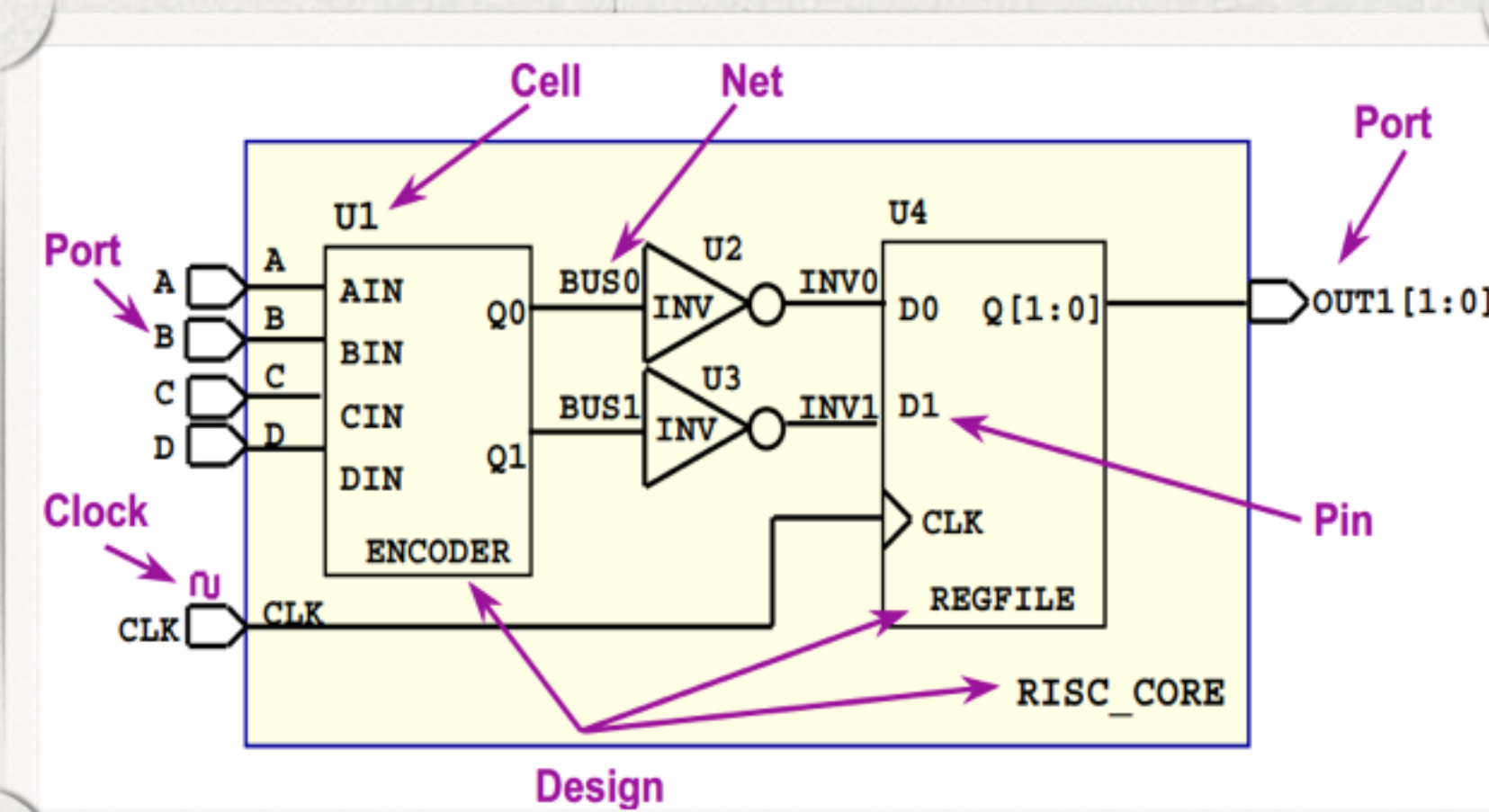
```
> get_pin -of [get_net INV1]
```

U3/Z

--cell object <-> pin object

```
>get_pins -of [get_cell U4]
```

{U4/D0 U4/D1 REGFILE/Q1 REGFILE/Q2}



综合软件与后端布局布线软件

