ClickNP: Highly Flexible and High Performance Network Processing with Reconfigurable Hardware

Bojie Li[§†] Kun Tan[†] Layong (Larry) Luo[‡] Yanqing Peng[•†] Renqian Luo[§†]
Ningyi Xu[†] Yongqiang Xiong[†] Peng Cheng[†] Enhong Chen[§]
[†]Microsoft Research [§]USTC [‡]Microsoft [•]SJTU

Microsoft Research

Challenge: FPGA programming

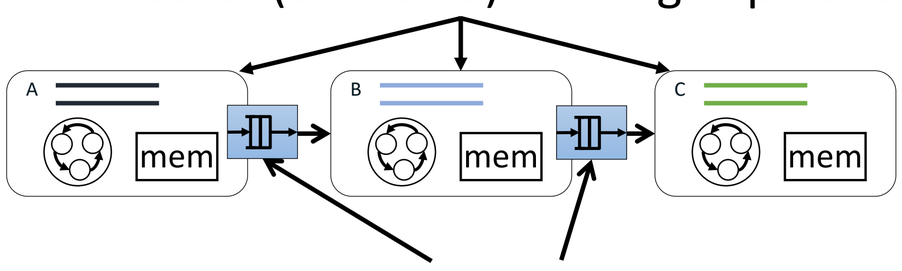
Hardware description languages: Verilog, VHDL…
Hard to program, hard to debug

ClickNP: Making FPGA accessible to software developers

- **Flexible**: high-level language
- **Modular**: familiar Click abstractions
- **High performance**
- **Joint CPU/FPGA packet processing**

Programming model

As if programming a **multi-core** processor
cores (*elements*) running in parallel



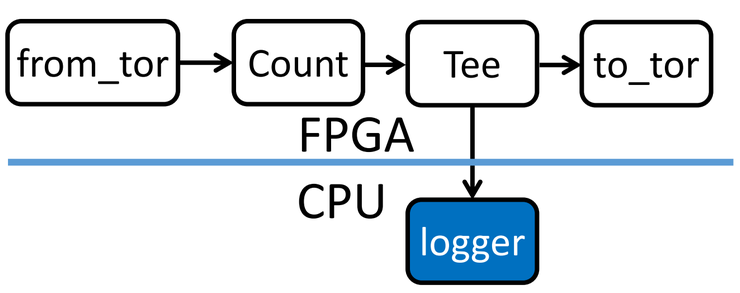
communicate via *channels*, not shared memory

ClickNP element

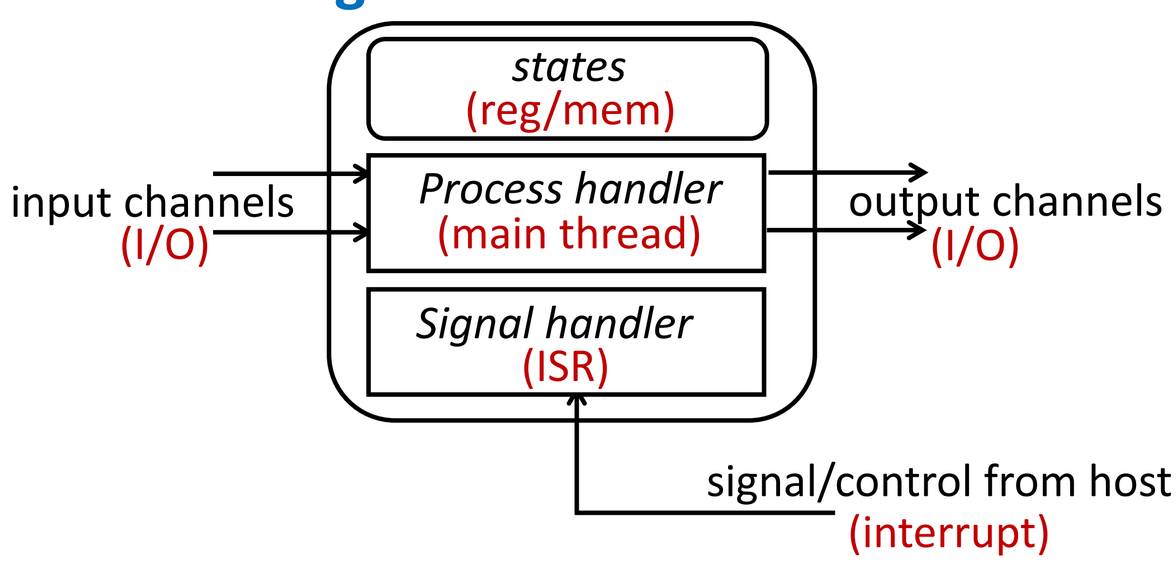
```
1  .element Count (flit -> flit) {
2    .state{
3      ulong count;
4    }
5    .init{
6      count = 0;
7    }
8    .handler{
9      if (test_input_port(PORT_1)) {
10       flit x;
11       x = read_input_port(PORT_1);
12       if (x.fd.sop)
13         count += 1;
14       set_output_port(PORT_1, x);
15     }
16   }
17   .signal{
18     ClSignal p;
19     p.Sig.LParam[0] = count;
20     set_signal(p);
21   }
22 }
```

ClickNP element graph

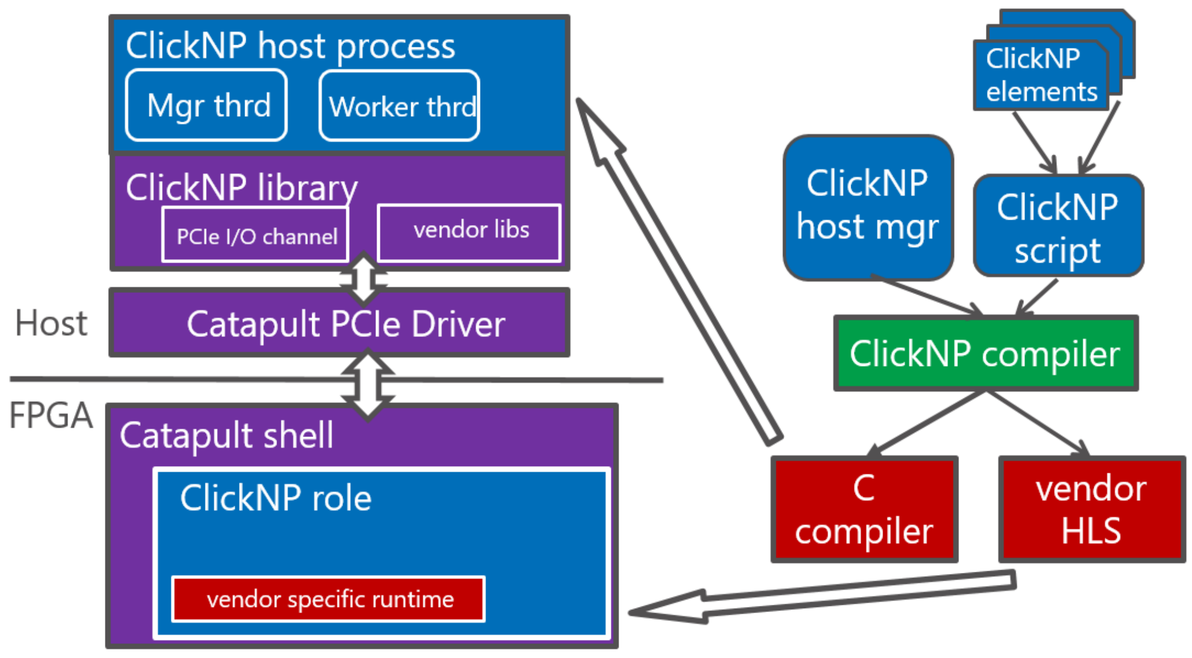
```
1 Count :: cnt @
2 Tee :: tee
3 host PktLogger :: logger
4
5 from_tor -> cnt -> tee [1] -> to_tor
6 tee [2] -> logger
```



Element: single-threaded core



Architecture



Demo: Stateful L4 load balancer