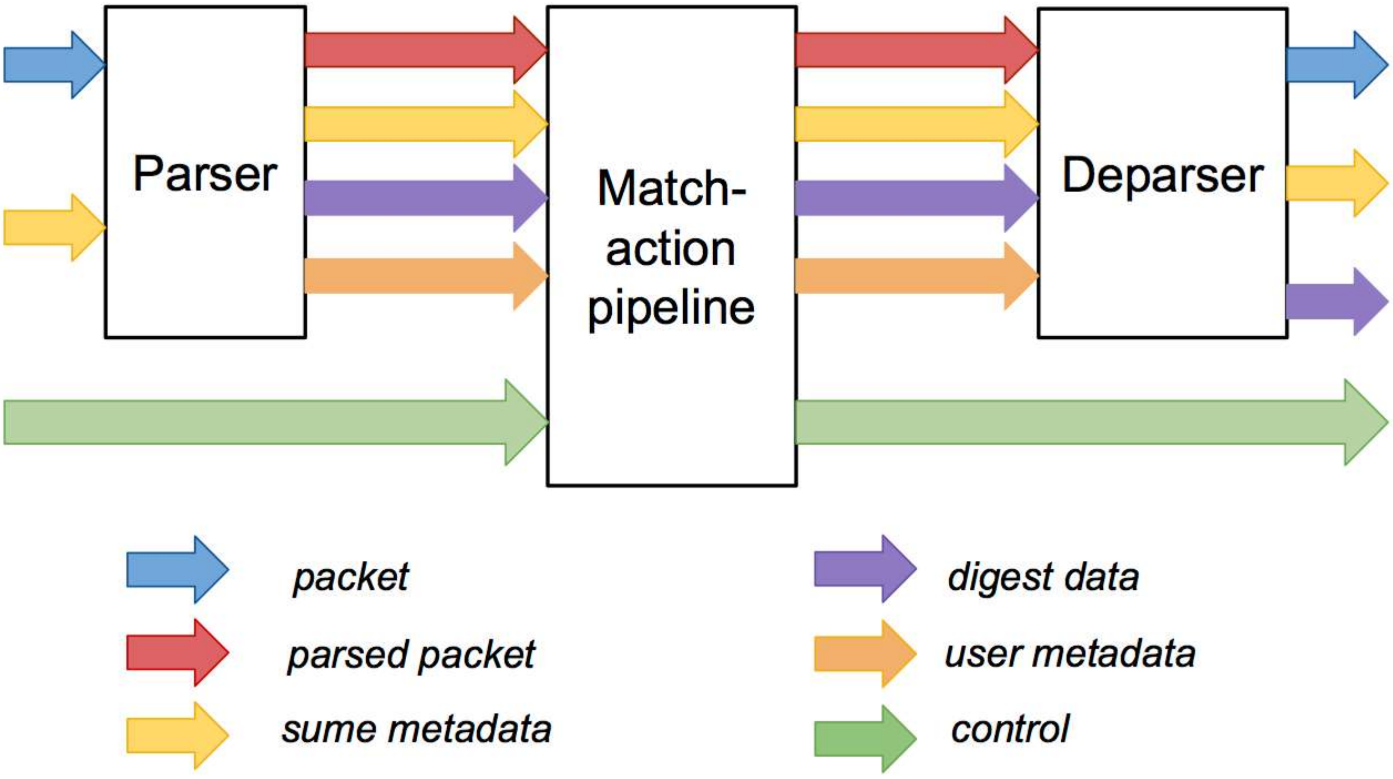
CS344

Build an Internet Router

SimpleSumeSwitch Architecture

We will be using the latest version of SimpleSumeSwitch architecture, which has not been fully merged into the upstream P4-NetFPGA-live repository. As such, the documentation of the SimpleSumeSwitch architecture on the P4-NetFPGA-public wiki is slightly out of date. Refer to this document instead for details about the operation of the SimpleSumeSwitch.

The SimpleSumeSwitch is the P4 architecture that is currently defined for the NetFPGA SUME. The architecture description can be found in /opt/Xilinx/SDNet//data/p4include/sume_switch.p4 or wherever you have installed Xilinx SDNet. The architecture consists of a single parser, single match-action pipeline, and single deparser. As shown below:



- **sume_metadata:** corresponds to the tuser bus in the SUME reference_switch design, it is defined as follows:

```
struct sume_metadata_t {
    bit<16> dma_q_size; // measured in 32-byte words
    bit<16> nf3_q_size; // measured in 32-byte words
    bit<16> nf2_q_size; // measured in 32-byte words
    bit<16> nf1_q_size; // measured in 32-byte words
    bit<16> nf0_q_size; // measured in 32-byte words
    bit<8> send_dig_to_cpu; // send digest_data to CPU
    bit<8> drop; // * DEPRECATED *
    port_t dst_port; // one-hot encoded (see below)
    port_t src_port; // one-hot encoded (see below)
    bit<16> pkt_len; // (bytes) unsigned int
}

The format of the dst_port and src_port fields is as follows:

  bit-7     bit-6     bit-5     bit-4     bit-3     bit-2     bit-1
(nf3_dma)-(nf3_phy)-(nf2_dma)-(nf2_phy)-(nf1_dma)-(nf1_phy)-(nf0_dma)-(
```

- `pkt_len` - the size of the packet (not including the Ethernet preamble or FCS) in bytes.
- `src_port` - the port on which the packet arrived. For example, if the packet arrived on port nf1 this field would be set to 0b00000100.
- `dst_port` - should be set by the user's P4 program to indicate which port or ports (if any) the packet should be sent out of. For example, to send a copy the packet out of both ports nf0 and nf2 this field should be set to 0b00010001.
- `drop` - this field is now deprecated and will be remove in a future version. To drop a packet set `dst_port` = 0.
- `send_dig_to_cpu` - The `digest_data` will always be prepended to the packet whenever it is forwarded over DMA to the CPU. However, if the P4 programmer *only* wants to send only the `digest_data` to the CPU and not the entire packet they should set the least significant bit of this field and ensure that all CPU port bits of the `dst_port` field are set to 0.

- `_q_size` - the size of each output queue, measured in terms of 32-byte words (rounded up). This is the size of the output queues when the packet *starts* being processed by the P4 program.

- **digest_data:** the format of this bus is defined by the P4 programmer. The only constraint is that it *must* be defined to be 256 bits wide. This data will be prepended to the packet when it is sent over DMA to the CPU.
- **user_metadata:** the format of this bus is also defined by the P4 programmer. It can be used to pass additional information between the parser, the M/A pipeline, and the deparser.
- **in/out control:** these signals are used to add/remove entries from tables and read/write control registers.

A Note About Interface Names

Many new users are often confused about the interface names. The NetFPGA SUME board has 4 SFP+ ports (a.k.a physical interfaces). We often call these interfaces `nf0`, `nf1`, `nf2`, and `nf3` where `nf0` is the port closest to the link lights on the SUME board. There is one bit in the `src_port` and `dst_port` fields for each of these ports (bits 0, 2, 4, and 6).

If you type `ifconfig` on the linux host machine after programming the FPGA you should something like the output shown below. The `nf0`, `nf1`, `nf2`, and `nf3` shown here mean something very different than what is explained above. These network interfaces are the means by which the host machine can communicate with the data-plane in the FPGA. There is also one bit for each of these interfaces in the `src_port` and `dst_port` fields (bits 1, 3, 5, and 7). So for example, if the data-plane wants to send a packet up to the host and have it arrive on the `nf0` linux network interface then it must set bit 1 of the `dst_port` field (e.g. `dst_port = 0b00000010`).

```
$ ifconfig
.
.
.
nf0       Link encap:Ethernet  HWaddr 02:53:55:4d:45:00
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:300 (300.0 B)
```

```
nf1       Link encap:Ethernet  HWaddr 02:53:55:4d:45:01
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:300 (300.0 B)

nf2       Link encap:Ethernet  HWaddr 02:53:55:4d:45:02
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:300 (300.0 B)

nf3       Link encap:Ethernet  HWaddr 02:53:55:4d:45:03
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:300 (300.0 B)
```