Parser Action:
One parse action is a 31-bit configuration and each user has 100 such parser actions
For each parse action:
[31:10] byte number from 0
[9:8] container type
[7:1] container idx
[0] validity bit

Packet Header Vector (PHV): 64 containers per type
6B * 64 + 4B * 64 + 2B * 64 + metadata
[6B 0, 6B 1, …, 6B 63] + [4B 0, 4B 1, …, 4B 63] + [2B 0, 2B 1, …, 2B 63] + [metadata (256-bit)]
As for metadata (keep it same as before for now):

- `[127:0]` is for the NetFPGA's `tuser` data.
- `[128]` is the discard flag where 1 means to drop the packet.
- `[140:129]` is the `VLAN ID`.
- `[255:141]` is reserved for other usage.

Key Extractor: match on 16 fields per type
The `key extractor` will extract the key out of PHV according to the instructions in the RAM. Specifically, in our design, we use the `VLAN ID` (i.e., `PHV[140:129]`) to index the instruction RAM in `key extractor` and then according to the instruction to extract the key from PHV.
Format:
6B: [container idx0, container idx 1, …, container idx 15] +
4B: [container idx0, container idx 1, …, container idx 15] +
2B: [container idx0, container idx 1, …, container idx 15] +
Keep all these parts same for now: [2-bit comp opcode] + [9-bit operand A] + [9-bit operand B]

Action engine:
64-bit per ALU, (64 * 3 + 1 = 193 ALUs in total). 8-bit for opcode, others can remain the same as the current setting.

New request (less priority):
1. Corresponding conf.txt-> axi stream generation must be changed
2. If-else statement in control flow
3. Ternary match