

Created By	Hemant Singh
------------	--------------

< Nested Bit-vector Structure and Flatten Midend Pass Support in p4c >

Reviewers

Department	Name/Title
	Mihai Bidiu, p4-dev

Modification History

Revision	Date	Originator	Comments
1	08/31/2018	HS	First draft
1.1	09/04/2018	HS	Updated requirements

Table of Contents

1	Purpose.....	3
1.1	Scope	3

1 Purpose

1.1 Scope

The document covers this PR: <https://github.com/p4lang/p4c/issues/562>

Specifically, p4c is modified to support:

1. Nested bit-vector structures.
2. Nested bit-vector structures in P4 header.
3. Casting of nested bit-vector structure to bit-field of same width or vice versa.
4. Item 1 and 2 as parameter in P4 Action, Control, Parser, and Table key.
5. The bmv2 backend changes to support items 1 and 2.
6. A P4 header_union with headers including nested bit-vector structure(s).
7. Only P4-16.

A nested header is not supported.

Another goal of the PR is to develop a midend pass to flatten any nested bit-vector structures.

Specifically, p4c is modified to support:

1. Sweeping over the whole P4 program and flattening any global and local nested bit-vector structures.
2. Sweeping over the whole P4 program to change use of any bit-vector structures to flattened form. An example use is an assignment statement, or a table key. During use, an issue exists, see the example code below.

```
ip = (bit<32>) b.sip; // b is a bit-vector struct with "sip_t sip" as a bit-vector struct for member. The struct sip has already been flattened before the above line of code is processed. Therefore, two assignment statements are generated with bit Slices used in LHS of the assignment statement.
```
3. If a parameter of a P4 Action, Control, and Parser is a nested bit-vector structure, the parameter flattened as a tuple akin to what is implemented in current `p4c/midend/nestedStructs.h[.cpp]`. Alternatively, the parameter could be left untouched (not flattened) and just the use of the parameter inside the Control block or Parser is flattened.