Formal Verification of *Bitwalker* with Frama-C

Jens Gerlach
Fraunhofer FOKUS, Berlin

Who is involved?

- Siemens provides implementation of bitwalker as part of modeling effort of subset 26-7
- CEA LIST provides Frama-C verification platform
- Fraunhofer FOKUS does the actual specification and verification

Bitwalker

- converts bit stream to/from integer
- used to fill ETCS data structures
- half a dozen small C functions with **`peek`**/ **`poke`** at its core
- Siemens implementation heavily relies on bit operators of C

Formal Specification

- use specification language ACSL of Frama-C for formal specification of
  - peek/poke (partially done)
  - bitwalker incremental
  - upper layer functions
- formal specifications have to be reviewed by Siemens

```
/*@
  requires 0 <= Startposition + Length <  UINT64_MAX;
  requires IsValidRange(Bitstream, BitstreamSizeInBytes);
  requires Length <= 64;
  assigns \nothing;

  behavior out_of_range:
    assumes OutOfRange(StreamIndex(Startposition +Length-1), BitstreamSizeInBytes);
    ensures \result == 0;

  behavior normal:
    assumes !OutOfRange(StreamIndex(Startposition+ Length-1),BitstreamSizeInBytes);
    ensures \result == BitSum(Startposition, Length, Bitstream);

  complete behaviors;
  disjoint behaviors;
*/
uint64_t Bitwalker_Peek(unsigned int Startposition,
                        unsigned int Length,
                        uint8_t Bitstream[],
                        unsigned int BitstreamSizeInBytes);
```

Formal Verification

- use Frama-C plugin WP for formal verification
- discuss with CEA LIST various strategies to deal with bit operations
  - special automatic theorem provers (Z3)
  - or interactive theorem prover (Coq)

Open Issues

- Which parts of the ACSL specification of *bitwalker* can be *generated* from higher level models?