EuroForm Workshop, Dresden, Germany

The MAXIM Example: Simulation and Verification using the ELLA Verification Environment

Howard Barringer
Graham Gough
Brian Monahan*
Alan Williams

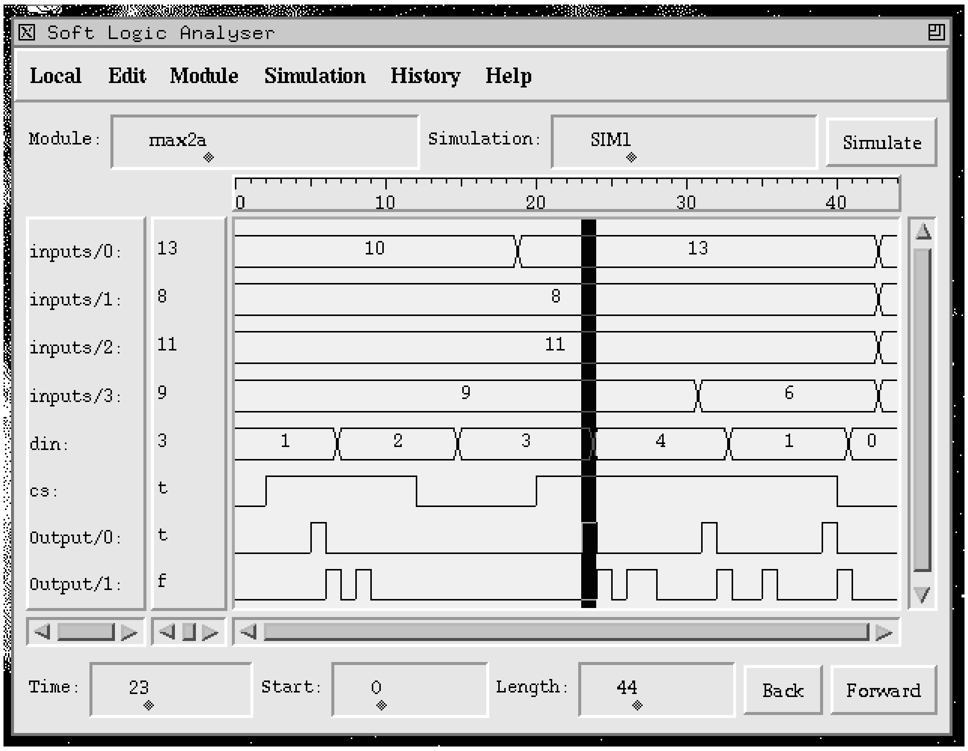
HaVe Group
Department of Computer Science
University of Manchester

funded by the UK SERC

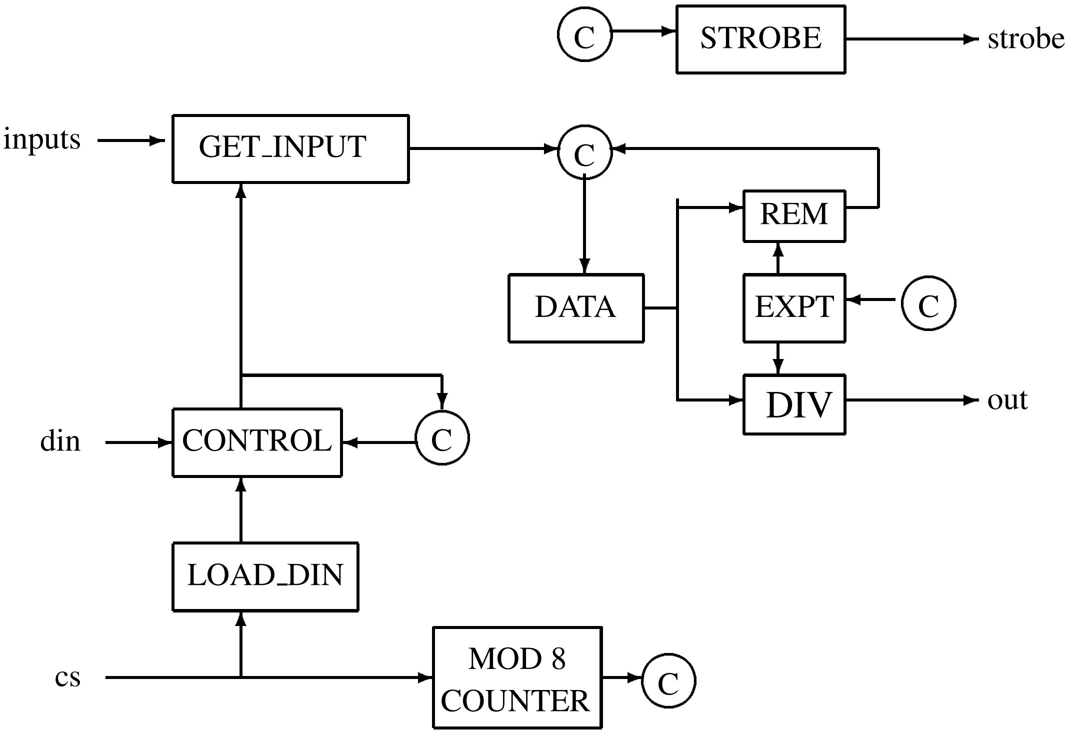
3 – 5 February 1995

*Now with Harlequin Ltd., Alderley Edge.

MAXIM: Simulation



MAXIM: Schematic



MAXIM: ELLA Description

```
TYPE bool    = NEW (f | t).     TYPE data    = NEW d/(0..15).
TYPE state   = NEW s/(0..7).    TYPE analog  = [4]data.
TYPE control = NEW c/(0..4).

FN DATA_STORE    = (data)    -> data:    IDELAY(d/0,1).
FN BOOL_STORE    = (bool)    -> bool:    IDELAY(f,1).
FN CONTROL_STORE = (control) -> control: IDELAY(c/0,1).
FN STATE_STORE   = (state)   -> state:   IDELAY(s/0,1).

FN INC_STATE = (state: x) -> state: ARITH (x+1) MOD 8.

FN LOAD_DIN = (state: st, bool: cs) -> bool:
    CASE (st,cs) OF
      (s/1, t): t
    ELSE f
    ESAC.

FN GET_INPUT = (analog: inputs, control: cntrl) -> data:
   CASE cntrl OF
     c/1 : inputs[1],     c/2 : inputs[2],
     c/3 : inputs[3],     c/4 : inputs[4]
   ELSE d/0
   ESAC.

FN INT_2_BOOL = (data: x)          -> bool: ARITH (1 + x).

FN REM2       = (data: x)          -> data: ARITH x MOD 2.
FN DIV2       = (data: x)          -> data: ARITH x % 2.
FN INT_REM    = (data: x, data: y) -> data: ARITH x MOD y.
FN INT_DIV    = (data: x, data: y) -> data: ARITH x % y.
FN SUB_FROM_7 = (state: x)         -> data: ARITH 7 - x.
FN EXPT       = (data:x)           -> data: ARITH 1 SL x.
```

MAXIM: ELLA Description (continued)

```
FN A_TO_D_1 =
    (analog: inputs, control: din, bool: cs) -> [2]bool:
BEGIN
  MAKE DATA_STORE    : data_store.
  MAKE CONTROL_STORE : control_store.
  MAKE STATE_STORE   : st.

  LET new_control_store = CASE LOAD_DIN(st,cs) OF
                            t : din
                          ELSE c/0
                          ESAC.
  LET new_st = CASE cs OF
                 t: INC_STATE(st)
               ELSE s/0
               ESAC.
  LET div_val = EXPT(SUB_FROM_7(st)).
  LET new_data_store = CASE st OF
                         (s/0: d/0 | s/1): d/0,
                         s/2: GET_INPUT(inputs,control_store),
                         s/3: data_store
                       ELSE INT_REM(data_store,div_val)
                       ESAC.
  LET out = CASE st OF
              (s/0 | s/1 | s/2 | s/3): f
            ELSE INT_2_BOOL(INT_DIV(data_store,div_val))
            ESAC.
  LET strobe = CASE st OF
                 s/3: t
               ELSE f
               ESAC.
  JOIN new_data_store -> data_store.
  JOIN new_control_store -> control_store.
  JOIN new_st -> st.
  OUTPUT (strobe,out)
END.
```

MAXIM: Verification

- A_TO_D_1 vs A_TO_D_1: all VCs proved in VCG

- A_TO_D_1 vs A_TO_D_2:
  - 61 VCs produced by VCG
  - most (all?) proved by inspection (e.g. contradictory constraints)
  - 6 proved using PVS, with additional axioms for mod int_to_digital.

- Proof complexity independent of number/size of analog inputs