Library Harmonization for Signal Integrity

Version	Date
0.0	2004/12/06

Table of Contents

1.0 Crosstalk noise

The magnitude of noise generated by crosstalk depends on the driver and receiver characteristics. Cell characterization for the driver requires hold resistance or I/V curve (more detailed) measured at the driving output pin, possibly state-dependent. Cell characterization for the receiver requires pin capacitance measured at the receiving input pin.

Cell characterization for noise detection involves noise-margin or noise-rejection. The criterion is whether noise at an input pin propagates to the output or not. Noise margin usually refers to a constant number, whereas noise rejection usually refers to the shape of the input waveform (pulse magnitude versus pulse width).

Cell characterization for noise propagation: output noise peak, output noise width, input-to-output delay of noise waveform dependent on input noise peak, input noise width, output load capacitance. Possibly, time-to-peak can be characterized as well.

1.1 Static current versus voltage

The I/V curve describes output current as a function of output voltage for one of the following steady-state conditions: output high, output low, or output tristate (if tristate pin).

Question: Why measure current in tristate mode? Shouldn't the current be zero by definition?

FIGURE 1. I/V curve description in Liberty

```
/* Liberty */
cell (CellName) {
  pin(PinName) {
    direction : tristate;
    timing() {
      /* lib_TimingModel for non-tristate arc */
      steady_state_current_high (TemplateReference)
        { values ( /* lib_Data */ ); }
      steady_state_current_low (TemplateReference)
        { values ( /* lib_Data */ ); }
    }
    timing() {
      /* lib_TimingModel for tristate arc */
      steady_state_current_tristate (TemplateReference)
        { values ( /* lib_Data */ ); }
    }
  }
}
```

Question: How is the positive sense of current measurement defined in Liberty?

Note: The independent variable is the voltage measured at the same pin as the current. The range is defined by the power/ground supply voltage.

Note: Arc-dependency can be inferred by the “related_pin” statement. State-dependency can be expressed using the “when” statement.

FIGURE 2. I/V curve description in ALF

```
/* ALF */
CELL CellName {
  PIN PinName {
    DIRECTION = output;
    ATTRIBUTE { tristate }
  }
  VECTOR ( ! PinName ) { // boolean expression for low
    CURRENT { PIN = PinName; FLOW = <in|out>; /* ALF data */ }
  }
  VECTOR ( PinName ) { // boolean expression for high
    CURRENT { PIN = PinName; FLOW = <in|out>; /* ALF data */ }
  }
  VECTOR ( PinName == 'bZ ) { // boolean expression for tristate
    CURRENT { PIN = PinName; FLOW = <in|out>; /* ALF data */ }
  }
}
```

Note: The FLOW annotation in ALF specifies the sense of current measurement (in = positive measurement into the pin).

Note: The logic state of the output pin is expressed in the boolean expression associated with the VECTOR. Optionally, a CHARACTERIZATION_CONDITION can be specified to define the logic state of the input pins used for the characterization measurement to drive the required logic state of the output pin.
Also, an optional EXISTENCE_CONDITION can be specified to define the necessary and sufficient set of logic states of the input pins that drive the required logic state of the output pin.

Note: More independent variables can be defined in addition to the voltage at the output pin: voltage at other pins, temperature, process condition ...

Note: Both arc-and state-dependency can be expressed using a more involved boolean expression associated with the VECTOR. In a steady-state scenario, the notion of an arc is somewhat artificial. For example, in the case of a 2-input NAND gate, there is one and only one condition that satisfies output low (both inputs high), but three conditions that satisfy output high (first input low, second input low, both inputs low). The conditions with one input low can be associated with the respective input-to-output arc, the condition with both inputs low cannot be associated with a particular arc.

1.2 Static hold resistance

The hold resistance represents dV/dI rather than I=f(V).

In Liberty, the dV/dI characteristic is reduced to four discrete numbers: dV/dI for high, low, above high, and below low. High and above high is for the logic high state of the output pin. The above high region is where the output voltage exceeds the positive power supply voltage. Low and below low is for the logic low state of the output pin. The below low region is entered when the output voltage exceeds the negative power supply voltage.

FIGURE 3. Hold resistance description in Liberty

```
/* Liberty */
cell (CellName) {
  pin(PinName) {
    direction : output;
    timing() {
      steady_state_resistance_high : number_for_high ; }
      steady_state_resistance_low : number_for_low ; }
      steady_state_resistance_above_high : number_for_above_high ; }
      steady_state_resistance_below_low : number_for_below_low ; }
    }
  }
}
```

In ALF, the logic state of the output is specified in the boolean expression associated with the output. The resistance dV/dI can be expressed with the same degree of detail as I=f(V), including state-dependency, independent variables etc.

FIGURE 4. Hold resistance description (general) in ALF

```
/* ALF */
CELL CellName {
  PIN PinName {
    DIRECTION = output;
  }
  VECTOR ( ! PinName ) { // boolean expression for low
    RESISTANCE { PIN = PinName; /* ALF data */ }
  }
  VECTOR ( PinName ) { // boolean expression for high
    RESISTANCE { PIN = PinName; /* ALF data */ }
  }
}
```

The ALF-equivalent to the four data points in Liberty would be a lookup table with two data points each, as shown below.

FIGURE 5. Hold resistance description (Liberty-equivalent) in ALF

```
/* ALF */
CONSTANT Vdd = number; // supply voltage
CELL CellName {
  PIN PinName {
    DIRECTION = output;
  }
  VECTOR ( PinName ) { // boolean expression for high
    RESISTANCE { PIN = PinName;
      HEADER {
        VOLTAGE { PIN = PinName; INTERPOLATION = floor;
          TABLE { 0 Vdd }
      } } TABLE { number_for_high number_for_above_high }
    }
  }
  VECTOR ( ! PinName ) { // boolean expression for low
    RESISTANCE { PIN = PinName;
      HEADER {
        VOLTAGE { PIN = PinName; INTERPOLATION = ceiling;
          TABLE { 0 Vdd }
      } } TABLE { number_for_below_low number_for_low }
    }
  }
}
```

Note: The annotation value “floor” specifies that the resistance for voltage “Vdd” and lower is “number_for_high”. The annotation value “ceiling” specifies that the resistance for voltage 0 and higher is “number_for_low”. That means, intermediate values are not interpolated.

1.3 Noise margin

1.4 Noise rejection

1.5 Noise propagation

2.0 Driver and receiver modeling

The purpose is to capture timing waveforms more accurately than with a single slewrate number. At the driver side, the transient voltage waveform, i.e. V= f(t) can be characterized. Alternatively, the transient current waveform, i.e., I=f(t) can be characterized.

At the receiver side, the voltage-dependency of the input pin capacitance can be characterized.

2.1 Transient waveform

Fill in material from ECSM and CCSM here.

In ALF, a transient waveform is associated with a VECTOR, wherein the vector expression generally defines a timing arc, optionally with state-dependency and/or with multiple switching input or output signals. The waveform data itself is described as a table with the independent variable time. Additional independent variables can be used, such as input slewrate, effective load capacitance, supply voltage, temperature, process condition etc. Note: To make the waveform easy to recognize it is recommended to use time as the innermost variable.

FIGURE 6. Waveform description in ALF with fixed time index

```
/* ALF */
  VECTOR ( ArbitraryVectorExpression ) {
    <CURRENT|VOLTAGE> { PIN = PinName; MEASUREMENT = transient;
      HEADER {
        TIME {
          FROM { PIN = RefPinName; EDGE_NUMBER = RefEdgeNumber; }
          INTERPOLATION = floor|ceiling|linear;
          TABLE { /* 10 numbers */ }
        }
        CAPACITANCE { PIN = PinName; INTERPOLATION = fit;
          TABLE { cap1 cap2 cap3 }
        }
      }
      TABLE {
        /* 10 numbers corresponding to cap1 */
        /* 10 numbers corresponding to cap2 */
        /* 10 numbers corresponding to cap3 */
      }
    }
  }
```

Note: ALF supports waveform descriptions of arbitrary quantities, as long as they can be defined with an ALF keyword (such as CURRENT, VOLTAGE etc.).

Note: The MEASUREMENT annotation value “transient” indicates that the model represents indeed a waveform (since the variable TIME could be used for another purpose as well, for example, expected life time of a device).

Note: The INTERPOLATION annotation should be used to indicate piece-wise constant or piece-wise linear waveform.

Note: For current waveforms, the FLOW annotation should be also included.

If additional independent variables are used, the model describes a set of waveforms rather than a single waveform. In this case it is often desirable to use different points in time to capture each waveform with a similar resolution. A “fast” waveform requires time points in a narrow range. A “slow” waveform requires time points in a wide range. This is supported in ALF by the use of a TEMPLATE and a template instantiation. The template instantiation is “dynamic” because the value of some placeholder is calculated dependent on the value of some other placeholder.

FIGURE 7. Waveform description in ALF with variable time index for variable current

```
TEMPLATE CurrentWaveform2D {
  CURRENT { PIN = <PinName>; MEASUREMENT = transient;
    HEADER {
      TIME {
        FROM { PIN = <RefPinName>; EDGE_NUMBER = <RefEdgeNumber>; }
        TABLE { <TimePoints> }
      }
      CAPACITANCE { PIN = <PinName>;
        TABLE { cap1 cap2 cap3 }
      }
    }
    TABLE { <Waveform1> <Waveform2> <Waveform3> }
  }
}
// later in the same library
VECTOR ( ArbitraryVectorExpression ) {
  CurrentWaveform2D = dynamic {
    PinName = PinName;
    RefPinName = RefPinName;
    RefEdgeNumber = RefEdgeNumber;
    Waveform1 {
      HEADER { TimePoints { TABLE { /* n1 numbers */ } } }
      TABLE { /* n1 numbers corresponding to cap1 */ }
    }
    Waveform2 {
      HEADER { TimePoints { TABLE { /* n2 numbers */ } } }
      TABLE { /* n2 numbers corresponding to cap2 */ }
    }
    Waveform3 {
      HEADER { TimePoints { TABLE { /* n3 numbers */ } } }
      TABLE { /* n3 numbers corresponding to cap3 */ }
    }
  }
}
```

A monotonic waveform can be described in a reverse way, i.e., the measurement describes the point in time where the quantity reaches a fixed value as opposed to the value of the quantity at a fixed point in time.

This scenario can also be described in a similar way as the previous one using a TEMPLATE and a template instantiation.

FIGURE 8. Waveform description in ALF with variable time index for fixed voltage crossings

```
TEMPLATE VoltageWaveformFixed2D {
  VOLTAGE { PIN = <PinName>; MEASUREMENT = transient;
    HEADER {
      TIME {
        FROM { PIN = <RefPinName>; EDGE_NUMBER = <RefEdgeNumber>; }
        TABLE { <Time1> <Time2> <Time3> <Time4> }
      }
      CAPACITANCE { PIN = <PinName>;
        TABLE { <CapacitancePoints> }
      }
    }
    TABLE { volt1 volt2 volt3 volt4 }
  }
}
// later in the same library
VECTOR ( ArbitraryVectorExpression ) {
  VoltageWaveformFixed2D = dynamic {
    PinName = PinName;
    RefPinName = RefPinName;
    RefEdgeNumber = RefEdgeNumber;
    Time1 {
      HEADER { CapacitancePoints { TABLE { /* n numbers */ } } }
      TABLE { /* n numbers */ }
    }
    Time2 {
      HEADER { CapacitancePoints { TABLE { /* n numbers */ } } }
      TABLE { /* n numbers */ }
    }
    Time3 {
      HEADER { CapacitancePoints { TABLE { /* n numbers */ } } }
      TABLE { /* n numbers */ }
    }
    Time4 {
      HEADER { CapacitancePoints { TABLE { /* n numbers */ } } }
      TABLE { /* n numbers */ }
    }
  }
}
```

2.2 Pin capacitance

Fill in material from ECSM and CCSM here.

A pin capacitance in ALF can be associated with a PIN. In this case, the capacitance can be one value or a pair of values, one for RISE and one for FALL. The value or the pair of values can be a function of variables associated with the same pin (such as the voltage on the pin) or of environmental variables (such as temperature or process).

Alternatively, the pin capacitance can be associated with a VECTOR. In this case, state- and arc-dependency can be modeled (by the vector expression), and the capacitance can be a function of arbitrary variables (e.g. voltage on another pin).

FIGURE 9. Pin capacitance description (general) in ALF

```
/* ALF */
CELL CellName {
  PIN PinName { DIRECTION = input;
    CAPACITANCE = number ;
  }
  PIN OtherPinName { DIRECTION = input;
    CAPACITANCE {
      RISE = numberForRise ;
      FALL {
        HEADER {
          PROCESS { TABLE { best worst } }
        } TABLE { numberForFallBest numberForFallWorst }
      }
      DEFAULT = numberForDefault ;
    }
  }
  VECTOR ( arbitraryVectorExpression ) {
    CAPACITANCE { PIN = YetAnotherPinName; /* ALF data */ }
  }
}
```

Multiple capacitance models for the same pin can be associated with multiple vectors. The vector expressions define mutually exclusive conditions for the usage of each particular model. The capacitance model that is directly associated with the pin shall be used per default. If both a RISE and a FALL model are directly associated with the pin, a DEFAULT value shall be associated with the pin as well.