scientific analog

Getting Started with *XMODEL*

Scientific Analog, Inc.

July 2017

Overview

- This material demonstrates how to launch *XMODEL* simulation and view its results using command-line commands (e.g., ***xmodel*** and ***xwave***)

- ***xmodel*** is a launcher script that executes the SystemVerilog simulator of your choice along with *XMODEL* libraries

- ***xwave*** is the waveform viewer optimized for viewing event-driven simulation results (*JEZ* format)

Get Ready

- Copy the tutorial package to your local directory:

```
$ cp -R ${XMODEL_HOME/tutorial/xmodel_basic ~
$ cd ~/xmodel_basic
```

A First Look on *XMODEL* Source

- sim/tb_stim/tb_stim.sv

```
`include "xmodel.h"

module tb_stim();
    xreal signal;
    sin_gen #(.delay(10e-9), .offset(0.5), .damp(0.05e9),
              .amp(1.0), .freq(0.5e9), .init_phase(0.5*M_PI))
            my_gen(signal);
    probe_xreal #(.format("jezascii")) my_probe(signal);
endmodule
```

Running *XMODEL*

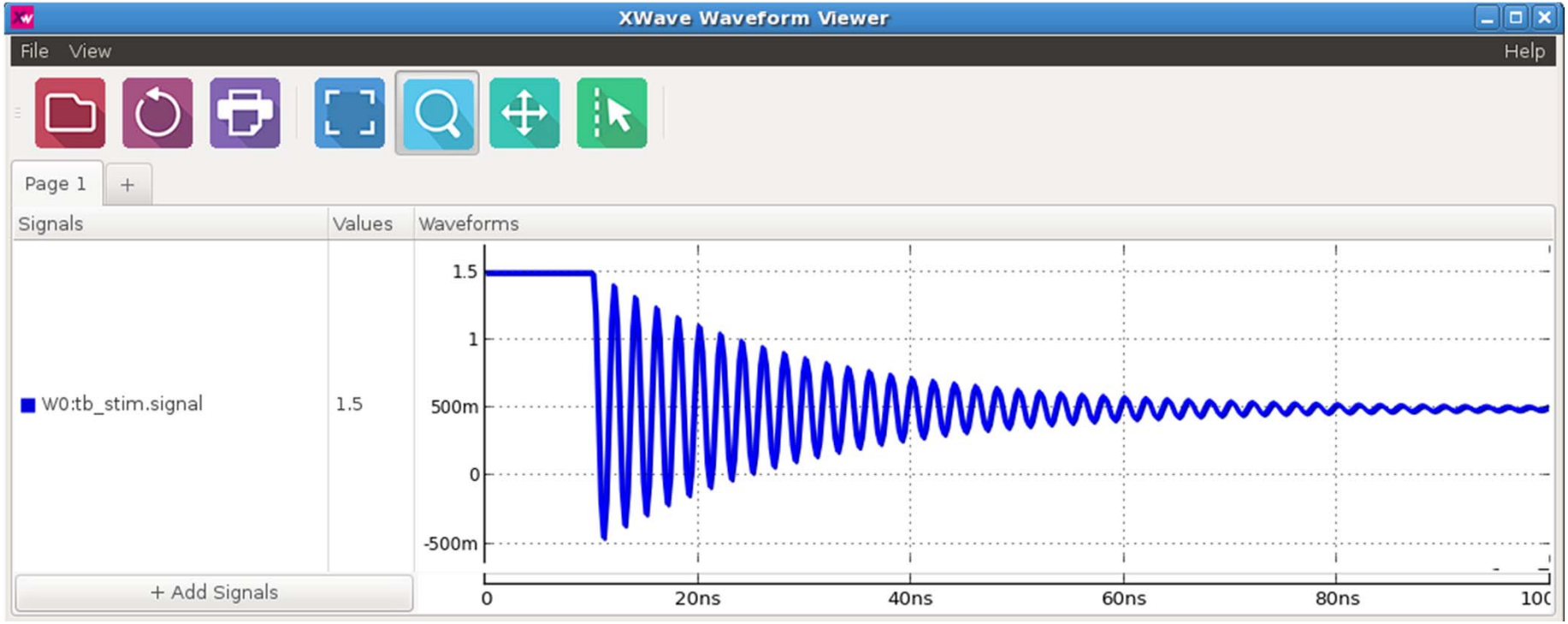
- On the linux prompt, run the simulation using the command '***xmodel***':

```
$ cd ~/training/XMODEL/sim/tb_stim
$ xmodel tb_stim.sv --top  tb_stim
```

- And view the result using '***xwave***':

```
$ xwave xmodel.jez -a
```

Waveform Results



- Can you guess what this example did?

Testbench *tb_stim.sv*

```
`include "xmodel.h"
```

Include XMODEL header file

```
module tb_stim();

      xreal signal;

      sin_gen #(.delay(10e-9), .offset(0.5), .damp(0.05e9),
                .amp(1.0), .freq(0.5e9), .init_phase(0.5*M_PI))
              my_gen(signal);

      probe_xreal #(.format("jezascii")) my_probe(signal);

endmodule
```

Testbench *tb_stim.sv* (2)

```
`include "xmodel.h"

module tb_stim();                                    Module declaration
      xreal signal;
      sin_gen #(.delay(10e-9), .offset(0.5), .damp(0.05e9),
                .amp(1.0), .freq(0.5e9), .init_phase(0.5*M_PI))
              my_gen(signal);
      probe_xreal #(.format("jezascii")) my_probe(signal);
endmodule
```

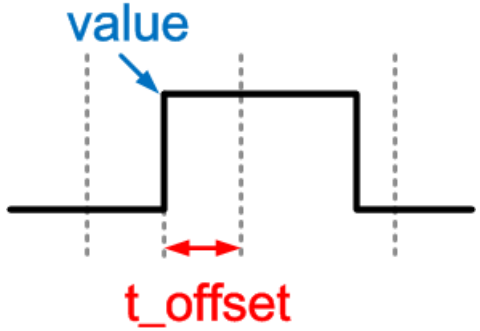
Testbench *tb_stim.sv* (3)

```
`include "xmodel.h"

module tb_stim();
      xreal signal;                                    Variable Declaration
      sin_gen #(.delay(10e-9), .offset(0.5), .damp(0.05e9),
                .amp(1.0), .freq(0.5e9), .init_phase(0.5*M_PI))
              my_gen(signal);
      probe_xreal #(.format("jezascii")) my_probe(signal);
endmodule
```

xbit and *xreal*

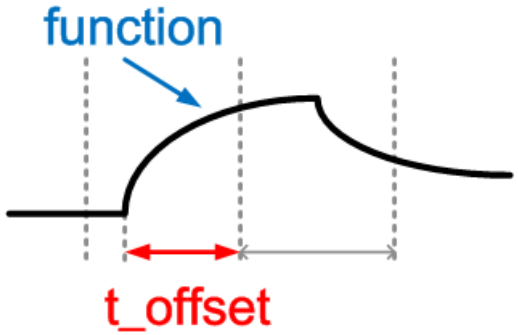
- ***xbit*** is for time-accurate digital signals

```
typedef struct {
    bit value;
    real t_offset;
} xbit;
```



- ***xreal*** is for continuous-time analog signals

```
typedef struct {
    chandle param_set;
    real t_offset;
    event flag;
} xreal;
```



Testbench *tb_stim.sv* (4)

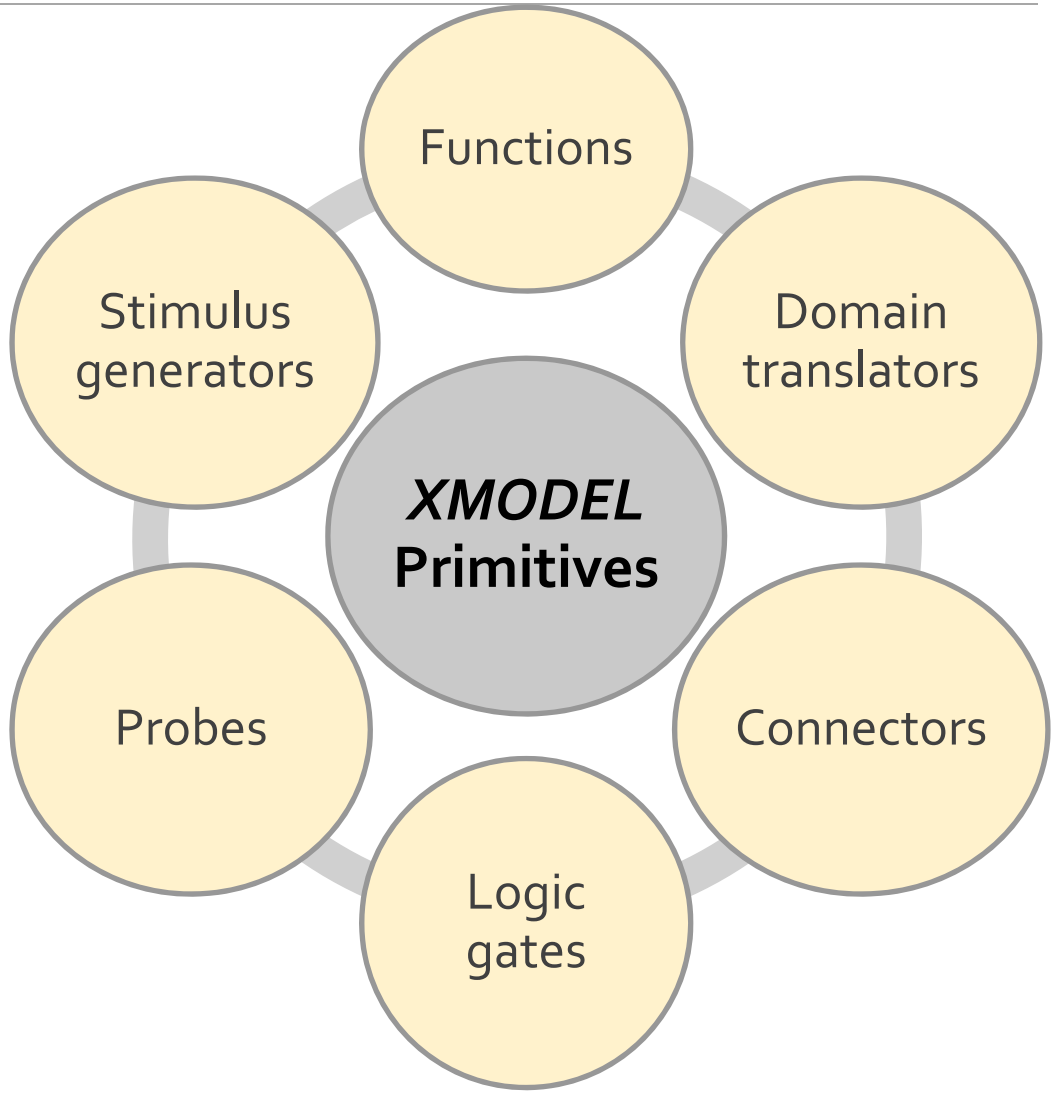
```
`include "xmodel.h"

module tb_stim();

      xreal signal;                                    XMODEL primitive instances

      sin_gen #(.delay(10e-9), .offset(0.5), .damp(0.05e9),
                .amp(1.0), .freq(0.5e9), .init_phase(0.5*M_PI))
              my_gen(signal);

      probe_xreal #(.format("jezascii")) my_probe(signal);

endmodule
```

- ***sin_gen*** generates a sinusoidal signal and ***probe_xreal*** records it into a file (by default, 'xmodel.jez')

XMODEL Primitives

- *XMODEL* package includes 6 different types of primitives

- Users can easily describe AMS systems without having to understand the *XMODEL* algorithms



XMODEL Primitives List

- Functions
  - add, scale, multiply, deriv, integ, integ_mod, filter, delay, select, limit, power, pwl_func, poly_func, transition, sample, compare, dac, adc, …
- Stimulus generators
  - dc_gen, noise_gen, step_gen, exp_gen, sin_gen, pwl_gen, clk_gen, pulse_gen, pat_gen, prbs_gen, …
- Logic gates
  - buf_xbit, inv_xbit, nand_xbit, nor_xbit, and_xbit, or_xbit, xor_xbit, xnor_xbit, mux_xbit, dff_xbit, …

XMODEL Primitives List (2)

- Domain translators
  - clk_to_freq, clk_to_phase, clk_to_period, clk_to_duty, clk_to_delay, freq_to_clk, phase_to_clk, period_to_clk, duty_to_clk, delay_to_clk
- Probes
  - probe_xbit, probe_xreal, probe_bit, probe_real, probe_freq, probe_phase, probe_period, probe_duty, probe_delay
- Connectors
  - xbit_to_bit, bit_to_xbit, xreal_to_real, real_to_xreal, xbit_to_xreal, xreal_to_xbit, …

sin_gen Primitive

- Generates a sinusoidal stimulus
- I/O description:

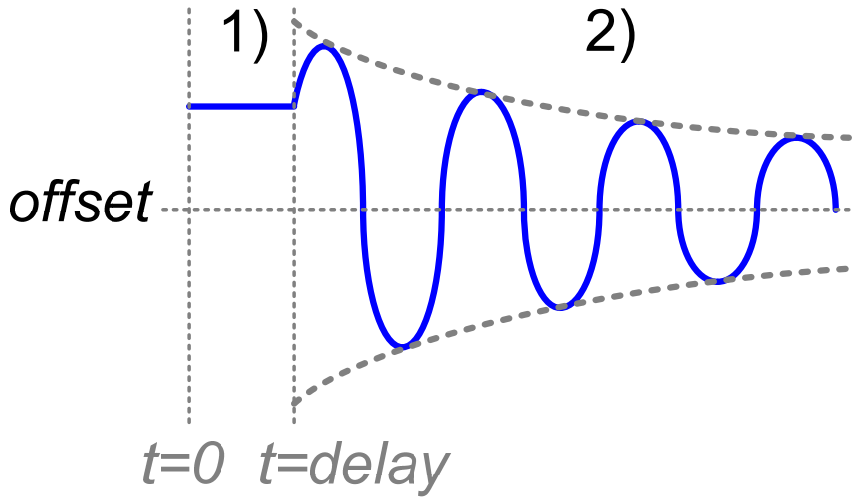
Name	I/O	Type	Description
out	output	xreal	signal output.

- Parameters:

Name	Type	Default	Description
offset	Real	0.0	offset.
amp	real	1.0	amplitude.
freq	real	1e9	frequency in Hz.
delay	real	0.0	initial delay in seconds.
damp	real	0.0	damping factor in 1/second.
init_phase	real	0.0	initial phase in radian.

sin_gen Primitive (2)

- More graphical illustration of parameters:



$$\begin{cases} 1)\ \text{offset}+\text{amp}\times\sin\left(\text{phase}\right) & \text{for } 0 \le t < \text{delay} \\ 2)\ \text{offset}+\text{amp}\times e^{-(t-\text{delay})\text{damp}}\times\sin\left[2\pi\text{freq}\left(t-\text{delay}\right)+\text{phase}\right] & \text{for delay} \le t \end{cases}$$

probe_xreal Primitive

- Records the waveform of an xreal-type signal
- Usage example:

```
probe_xreal  #(.start(1e-9), .filename("mywave.jez"))
             probe(in);
```

- Parameters:

Name	Type	Default	Description
filename	string	xmodel.jez	output filename.
start	real	0.0	abs. time to start the recording in seconds.
stop	real	-1.0	abs. time to stop the recording in seconds.
format	string	jezbinary	format version.

Accessing On-line Documentation

- Use '-h' command for on-line help:

```
$ xmodel  -h
…
list of help topics:
    function    Functions
    gate        Logic gates
    circuit     Circuit elements
    stim        Stimulus generators
    meas        Probes
    vdt         Domain translators
    connect     Connectors
```

Accessing On-line Documentation (2)

- Use '-h TOPIC' to get a list of primitives of each category:

```
$ xmodel -h stim
================================
TOPIC stim
================================
The XMODEL stimulus generator primitives provide means to
generate various stimulus waveforms both in analog and
digital format.

list of stimulus generator primitives:
    clk_gen             A digital clock generator.
    dc_gen              Analog DC generator
    exp_gen             Analog exponential signal generator
    noise_gen           Noise generator
    …
```

Accessing On-line Documentation (3)

- Use '-h PRIMITIVE' to get the documentation on each primitive:

```
$ xmodel -h sin_gen
==============================
PRIMITIVE sin_gen
==============================
Analog sinusoid generator

The 'sin_gen' primitive generates a sinusoidal signal that
can optionally be exponentially decaying or frequency/
amplitude-modulated.

The generated stimulus waveform V(t) is defined as follows:
    for t < delay:
        V(t) = offset + amp*AM_offset*sin(init_phase)
…
```

Running *XMODEL* Simulation

- Basic command to launch *XMODEL* simulation is:

```
$ xmodel file1.sv file2.sv … --top top_cell
```

  - top_cell is the name of the top-level module
- More available options include:
  - **--simtime** : specifies the simulation time
  - **--timescale** : specifies Verilog time scale and precision
  - **--simulator** : specifies the simulator (vcs,modelsim,ncverilog)
  - **--stat** : enables statistical simulation
  - **--clean** : clean up the simulation result files
- See the complete list by typing "xmodel –h"

Using Makefile for Batch Processing

- By writing the commands in a Makefile, you can repeat them simply by typing "make"
- Example: sim/tb_stim/Makefile

```
# Makefile ----------------------------------------
all: runsim plotwave
runsim:
    xmodel tb_stim.sv --top tb_stim
plotwave:
    xwave -a xmodel.jez
clean:
    xmodel --clean
```

- To launch the simulation and clean up:

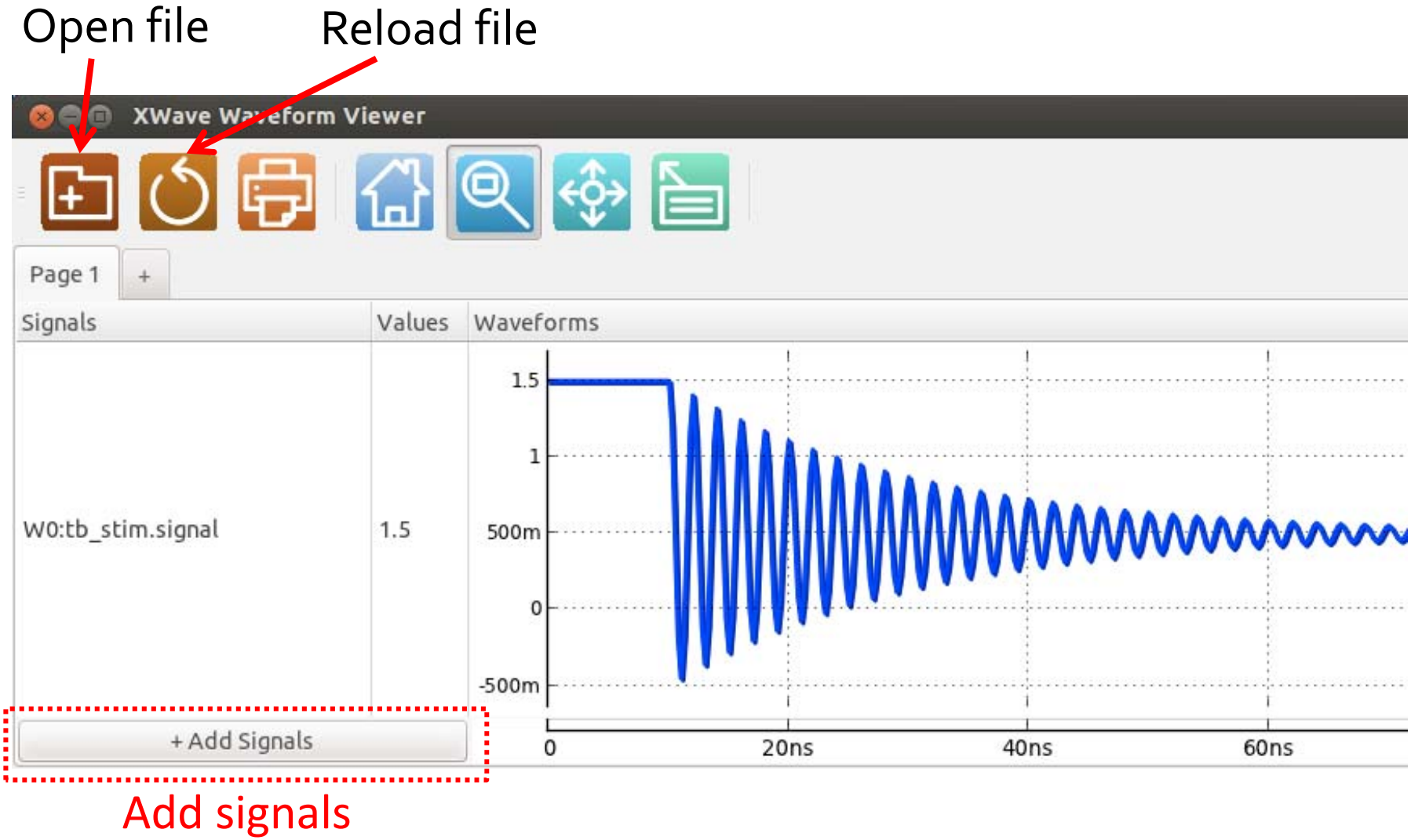
```
$ make
$ make clean
```

XWAVE Waveform Viewer

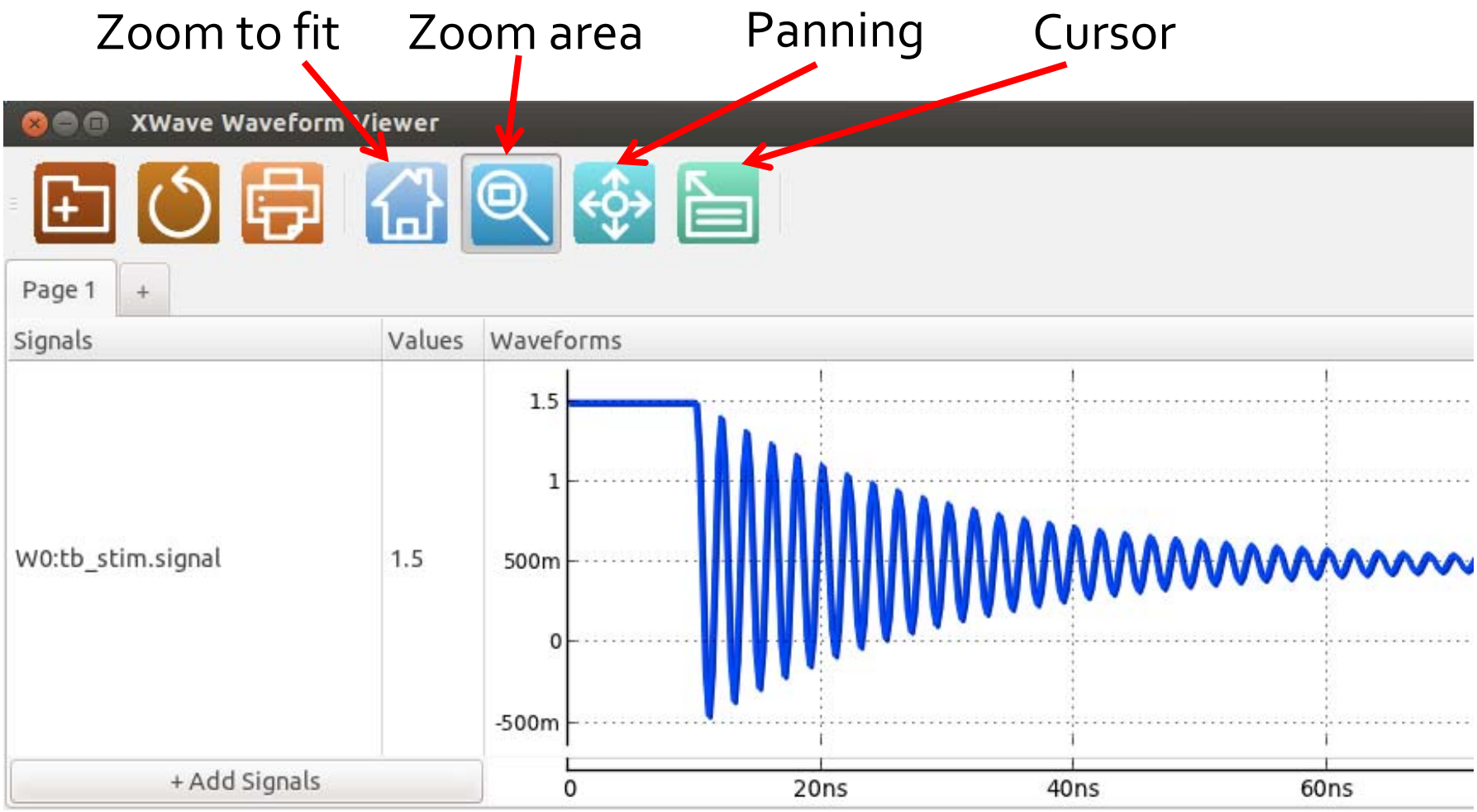
- Usage: **xwave** [options] [filename]
  - For example: `xwave -a xmodel.jez`

- Available options (type '`xwave -h`' for a full list):
  - -a, --all : display all waveforms.
  - -l, --list : display the list of variables.
  - -b, --xbit, --bit : display all xbit/bit waveforms.
  - -r, --xreal, --real : display all xreal/real waveforms.

XWAVE Commands



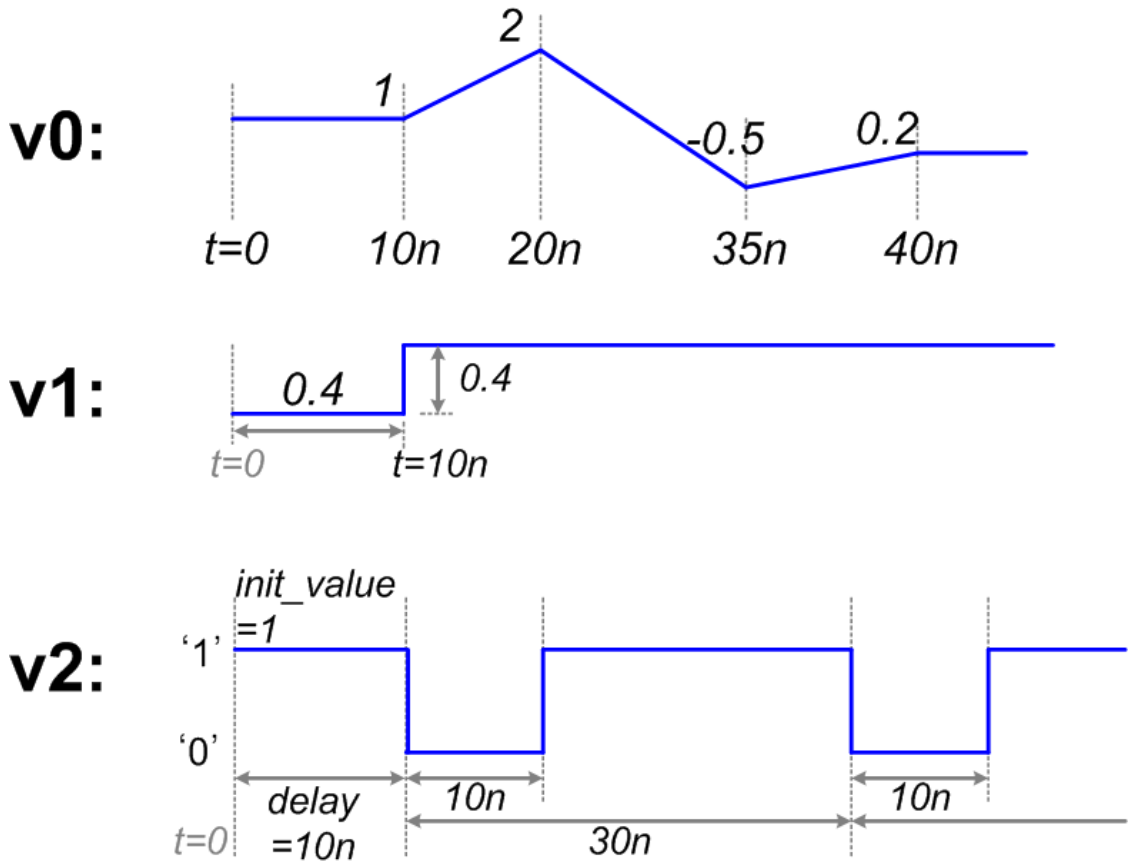
XWAVE Commands (2)



- Type 'u' to restore previous zoom settings

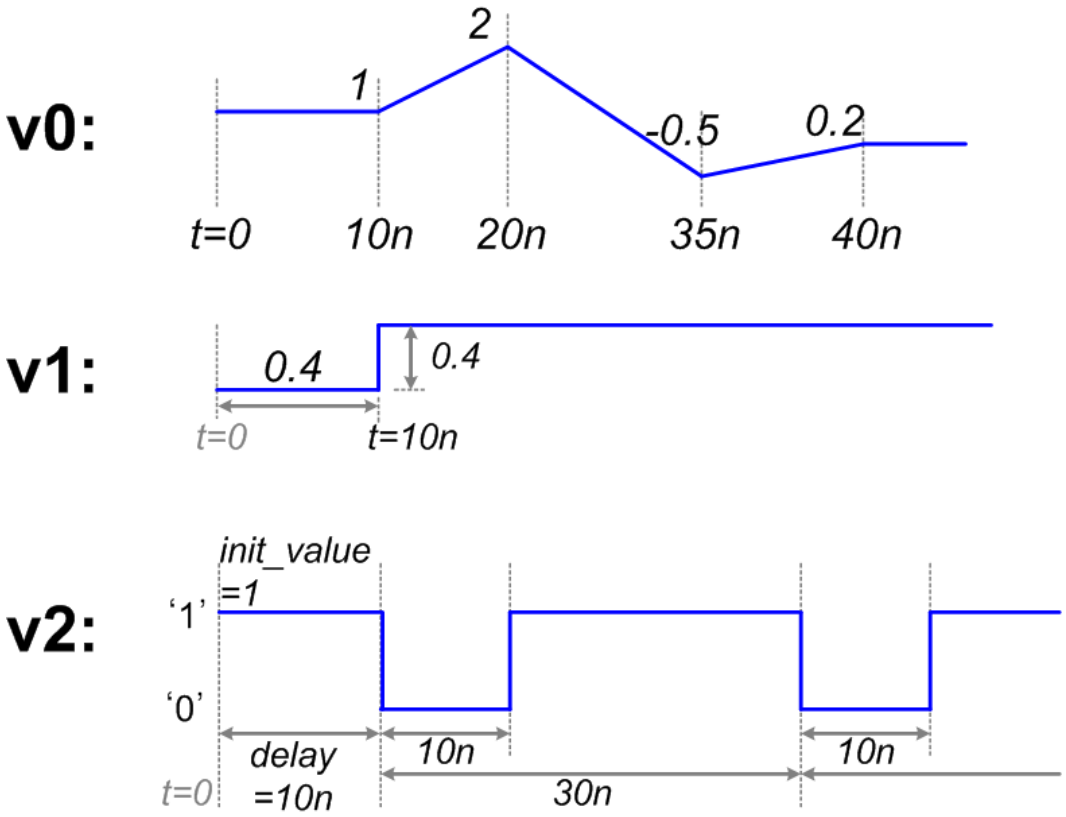
Exercise

- Write a testbench that generates signals v0, v1, v2



The skeleton code can be found in: XMODEL/sim/ tb_stim_ex/ tb_stim.sv

Exercise Hints



Use:

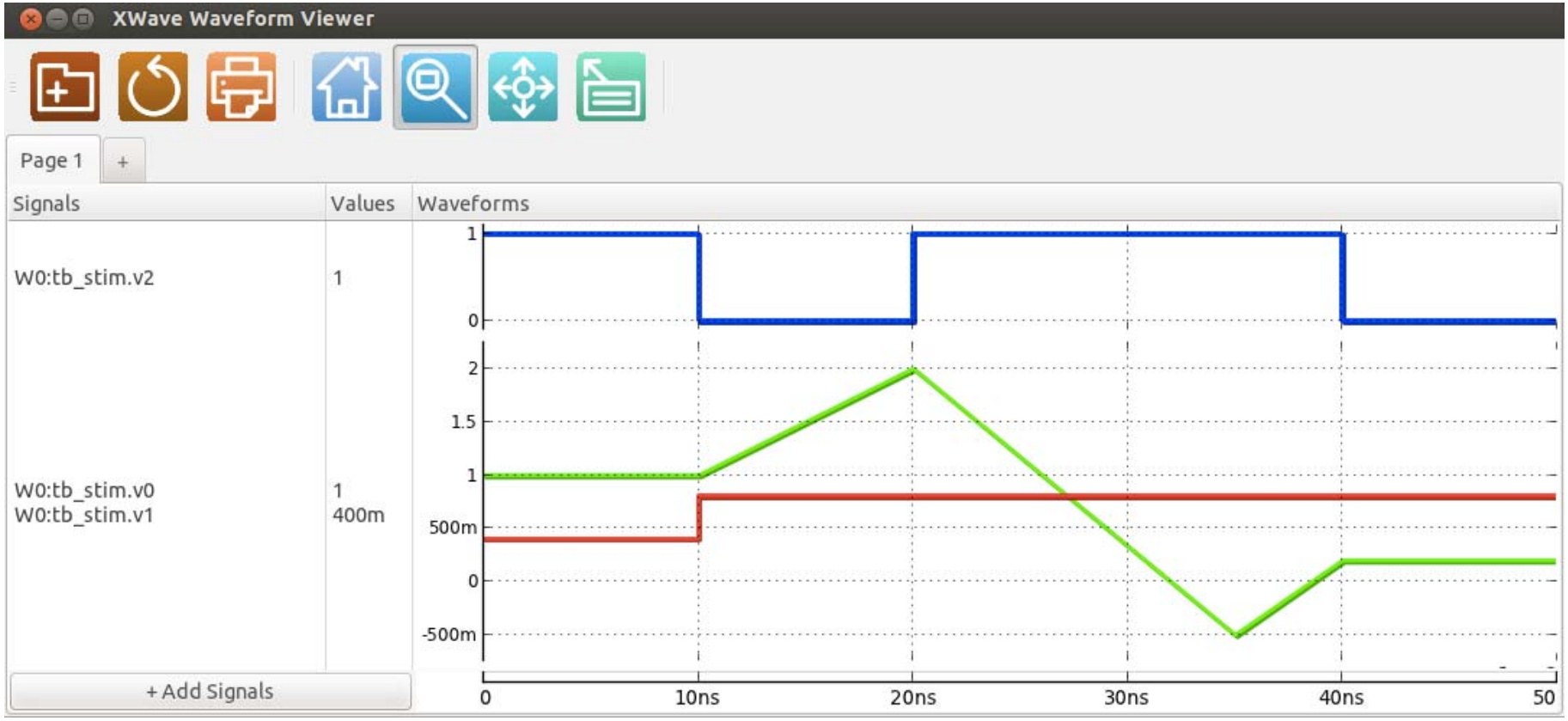
- ***pwl_gen*** primitive (piece-wise linear)
- ***step_gen*** primitive
- ***pulse_gen*** primitive

Answer (Only the Body Part Shown)

```
// variables
xreal v0;
xreal v1;
xbit  v2;

// stimulus instances
pwl_gen    #(.data('{10e-9, 1, 20e-9, 2, 35e-9, -0.5, 40e-9, 0.2}))
           v0_gen(v0);
step_gen   #(.init_value(0.4), .change(0.4), .delay(10e-9))
           v1_gen(v1);
pulse_gen  #(.init_value(1), .delay(10e-9), .width(10e-9), .period(30e-9))
           v2_gen(v2);

// probe instances
probe_xreal    probe_v0(v0);
probe_xreal    probe_v1(v1);
probe_xbit     probe_v2(v2);
```

Waveform Results:

- **xwave –a xmodel.jez**



- Note: try to move/merge/split the waveforms