Elec 422: VLSI Design I

Additional Information on Crystal

Fall 1998, October 15, 1998

1 Crystal Overview

Examples for crystal are located in ~/home/cavallar/crystal.

- add.sim: .sim file for the full adder cell.
- add.cryvec: test file which can be read by the "source" command.
- add.crystalresults: a sample session with crystal.

Refer also to the man page and to the tutorial at the end of the Elec 422 manual for more information on crystal.

Note that crystal can create an overlay file for *magic* similar to the design rule feedback areas. This is described in Section 8.1 of "Using Crystal for Timing Analysis" in the Elec 422 manual. There are a number of options that need to be set for the feedback areas to align with the *magic* files.

- Within *magic*, you need to set the extraction style to get the correct λ scale. Do this with:

  ```
  :extract style lambda=1.0
  ```

  then continue with circuit extraction and *ext2sim*.
- Within *crystal*, then enter:

  ```
  options graphics magic units 1.0
  ```

  to tell *crystal* about the units and graphics style.
- Also note that *crystal* has a "source" command to read in command files that can contain "options" statements.

2 Full Adder Cell Example

In this example a full adder cell made up of exclusive OR circuits and NAND gates will be used. The block diagram shows a bit-slice structure, and the magic layout shows the details of the various cells.

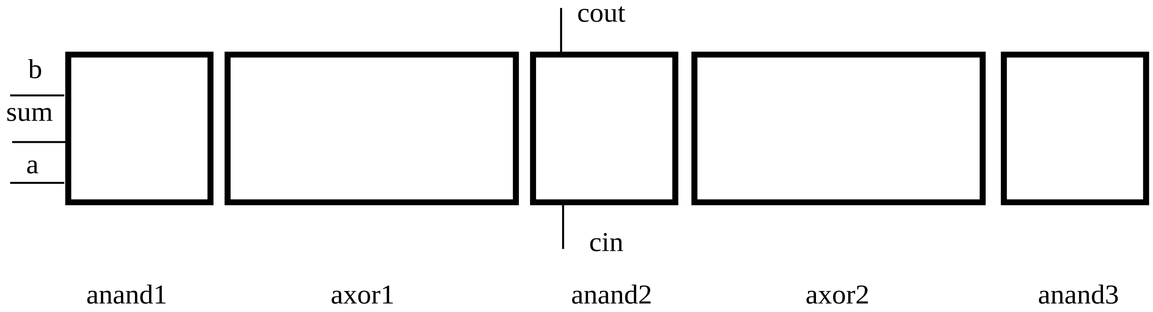

Figure 1: Full adder circuit block diagram.

2.1 Input Files for Crystal

The input files for *crystal* include the standard .sim transistor netlist presented below.

```
| units: 100 tech: scmos
p 48_0/a cout Vdd 2 6 523 32
p 46_0/out cout Vdd 2 6 536 32
n 48_0/a cout 48_0/6_86_50# 2 3 528 9
n 46_0/out 48_0/6_86_50# GND 2 3 533 9
p cin 47_0/abar Vdd 2 6 448 32
p cin 47_0/6_18_54# Vdd 2 6 458 32
p 46_0/b 47_0/6_18_54# Vdd 2 6 471 32
p 46_0/b 47_0/bbar Vdd 2 6 505 32
p 47_0/abar sum 47_0/6_18_54# 2 6 481 27
p 47_0/bbar sum 47_0/6_18_54# 2 6 495 27
n cin GND 47_0/abar 2 3 447 8
n cin GND 47_0/6_32_16# 2 3 463 8
n 46_0/b 47_0/6_32_16# sum 2 3 468 8
n 47_0/abar sum 47_0/6_78_16# 2 3 486 8
n 47_0/bbar 47_0/6_78_16# GND 2 3 494 8
n 46_0/b 47_0/bbar GND 2 3 509 8
p cin 46_0/out Vdd 2 6 418 32
p 46_0/b 46_0/out Vdd 2 6 432 32
n cin 46_0/out 46_0/6_86_50# 2 3 424 9
n 46_0/b 46_0/6_86_50# GND 2 3 429 9
p a 45_0/abar Vdd 2 6 344 32
p a 45_0/6_14_54# Vdd 2 6 354 32
p b 45_0/6_14_54# Vdd 2 6 367 32
p b 45_0/bbar Vdd 2 6 399 32
p 45_0/abar 46_0/b 45_0/6_14_54# 2 6 377 27
p 45_0/bbar 46_0/b 45_0/6_14_54# 2 6 389 27
n a GND 45_0/abar 2 3 346 8
n a GND 45_0/6_28_16# 2 3 359 8
n b 45_0/6_28_16# 46_0/b 2 3 364 8
```

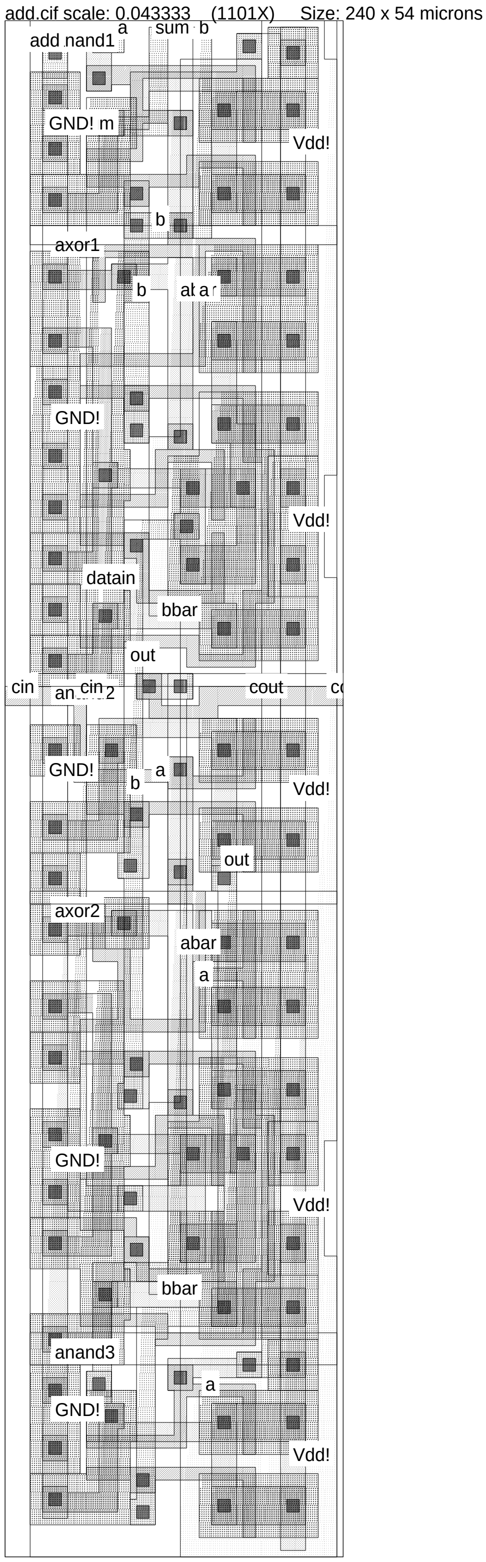

Figure 2: Full adder circuit layout.

```
n 45_0/abar 46_0/b 45_0/6_72_16# 2 3 381 8
n 45_0/bbar 45_0/6_72_16# GND 2 3 387 8
n b 45_0/bbar GND 2 3 403 8
p a 48_0/a Vdd 2 6 318 32
p b 48_0/a Vdd 2 6 331 32
n a 48_0/a 44_0/6_94_50# 2 3 327 9
n b 44_0/6_94_50# GND 2 3 331 9
C 45_0/bbar GND 38
C 45_0/6_14_54# GND 67
C 45_0/abar GND 39
C a GND 35
C b GND 35
C 47_0/6_78_16# GND 11
C sum GND 72
C 47_0/bbar GND 37
C 47_0/6_18_54# GND 70
C 47_0/abar GND 39
C cin GND 36
C 46_0/b GND 94
C cout GND 59
C 46_0/out GND 56
C 48_0/a GND 63
C Vdd GND 320
```

In addition, a crystal vector file is needed. The following example sets the graphics options, marks the inputs and outputs of the circuit, set a delay statement, and starts the critical path analysis.

```
options graphics magic units 1.0

inputs a b cin
outputs cout sum
delay a -1 0
critical
```

2.2 Output from Crystal

The following is a typescript from running *crystal* saved with the UNIX *script* command. After running "source add.cryvec" to build the circuit and perform the initial critical path analysis, several commands were entered interactively.

```
Script started on Tue Oct 25 10:48:23 1994
george% crystal add.sim
Crystal, v.2
: build add.sim
[0:00.0u 0:00.1s 18k]
```

```
: source add.cryvec
: options graphics magic units 1.0
[0:00.0u 0:00.0s 27k]
: inputs a b cin
[0:00.0u 0:00.0s 27k]
: outputs cout sum
[0:00.0u 0:00.0s 27k]
: delay a -1 0
Marking transistor flow...
Setting Vdd to 1...
Setting GND to 0...
(21 stages examined.)
[0:00.0u 0:00.0s 28k]
: critical
Node cout is driven high at 7.49ns
      ...through fet at (536, 32) to Vdd after
    46_0/out is driven low at 6.65ns
      ...through fet at (424, 9) to 46_0/6_86_50#
      ...through fet at (429, 9) to GND after
    46_0/b is driven high at 4.66ns
      ...through fet at (389, 27) to 45_0/6_14_54#
      ...through fet at (354, 32) to Vdd after
    a is driven low at 0.00ns
[0:00.0u 0:00.0s 28k]
[0:00.0u 0:00.0s 28k]
: critical -g add.crymag -s add.cryspice
[0:00.0u 0:00.0s 37k]
: critical 2
Node sum is driven high at 7.45ns
      ...through fet at (495, 27) to 47_0/6_18_54#
      ...through fet at (471, 32) to Vdd after
    46_0/b is driven low at 4.27ns
      ...through fet at (381, 8) to 45_0/6_72_16#
      ...through fet at (387, 8) to GND after
    45_0/abar is driven high at 0.71ns
      ...through fet at (344, 32) to Vdd after
    a is driven low at 0.00ns
[0:00.0u 0:00.0s 37k]
: critical -g add.cry2mag -s add.cry2spice 2
[0:00.0u 0:00.0s 37k]
: quit
[0:00.1u 0:00.2s 37k] Crystal done.
george% exit
george%
script done on Tue Oct 25 10:54:43 1994
```

The above ": critical -g add.crymag -s add.cryspice" command produced a *magic* feedback file and a file for further *spice* analysis. The "add.crymag" file contains commands to mark the nodes in the *magic* layout window. From within *magic*, type ":source add.crymag" and white feedback paint will appear. It is helpful to zoom in on the feedback areas and note the transistors that are selected. From the selected transistors, you can get a sense of the output node that is in the critical path.

```
box 536 32 538 34
feedback add "[3]7.5ns,rise"
box 424 9 426 11
feedback add "[2]6.7ns,fall"
box 429 9 431 11
feedback add "[2]6.7ns,fall"
box 389 27 391 29
feedback add "[1]4.7ns,rise"
box 354 32 356 34
feedback add "[1]4.7ns,rise"
echo "Crystal information entered as feedback."
```

The *spice* file can be used later to get more detailed analysis of the critical path.

```
* Nodes 4-5 correspond to cout (see source at 523,32)
* Nodes 6-7 correspond to 48_0/6_86_50# (see drain at 528,9)
* Nodes 8-9 correspond to 46_0/out (see gate at 533,9)
* Nodes 10-11 correspond to 46_0/6_86_50# (see drain at 424,9)
* Nodes 12-13 correspond to 46_0/b (see gate at 429,9)
* Nodes 14-15 correspond to 45_0/6_14_54# (see source at 354,32)
* Nodes 16-17 correspond to 45_0/6_28_16# (see drain at 359,8)
* Nodes 18-19 correspond to a (see gate at 327,9)

C1 13 0 0.120pf
C2 15 0 0.070pf
C3 5 0 0.060pf
C4 9 0 0.058pf
M1 1 9 5 1 p l=2.0u w=6.0u
M2 0 9 7 0 n l=2.0u w=3.0u
M3 7 1 5 0 n l=2.0u w=3.0u
M4 11 1 9 0 n l=2.0u w=3.0u
M5 0 13 11 0 n l=2.0u w=3.0u
M6 1 13 9 1 p l=2.0u w=6.0u
M7 15 0 13 1 p l=2.0u w=6.0u
M8 1 19 15 1 p l=2.0u w=6.0u
M9 17 19 0 0 n l=2.0u w=3.0u
M10 13 1 17 0 n l=2.0u w=3.0u

* Initial conditions:
.ic v(17)=0.000000
.ic v(11)=5.000000
```

```
.ic v(7)=0.000000
.ic v(13)=0.000000
.ic v(15)=0.000000
.ic v(5)=0.000000
.ic v(9)=5.000000

vdd 1 0 5.0
vin 19 0 pulse(5 0 0ns 0ns 0ns)
.tran 0.05ns 10ns
.plot tran V(5) (0,5)
.end
```

In the following figure, I have marked the nodes that crystal highlighted in the text output and the *magic* feedback output. On the circuit diagram of the full adder, the critical path is also drawn.

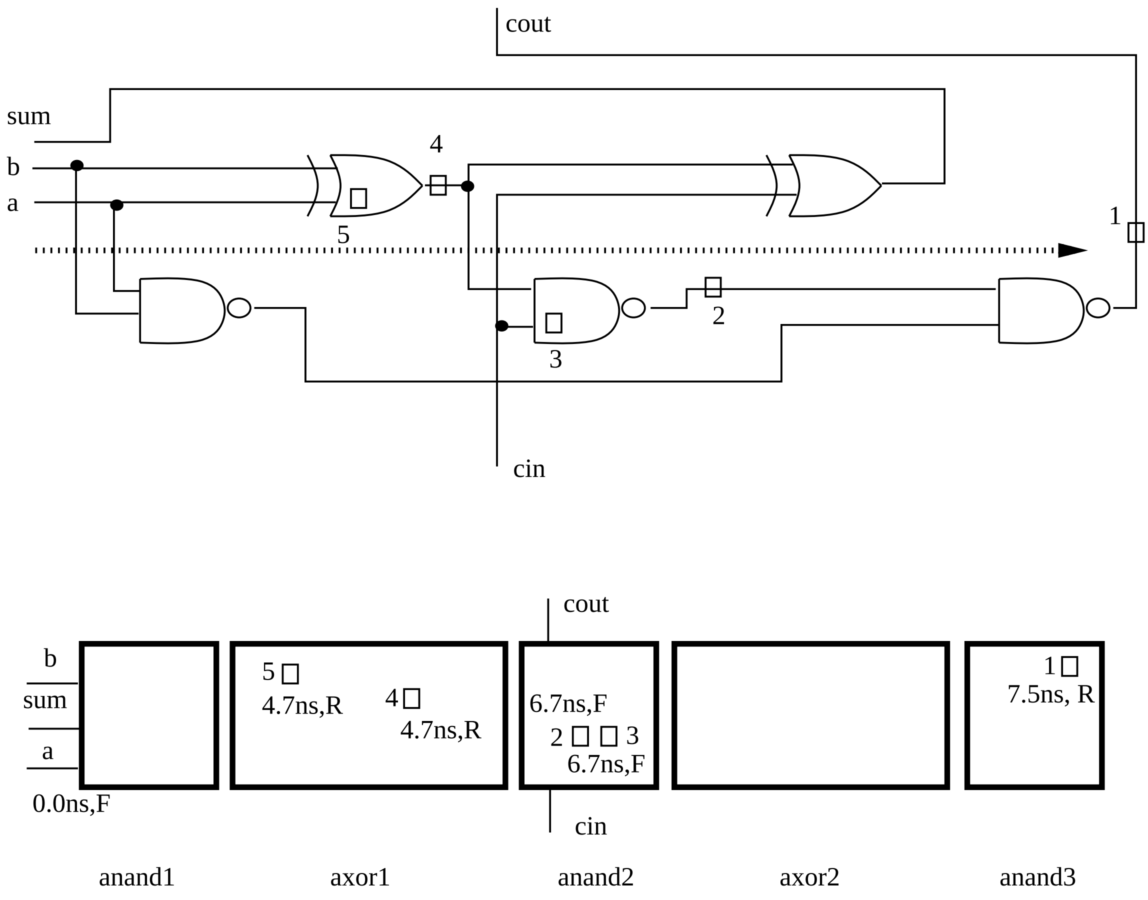

Figure 3: Full adder circuit.