B.1 List of Examples

These synthesis results for all examples can be easily reproduced by using the scripts `qvhdl.tcl` in the VHDL or Verilog directories of the CD-ROM. Run the TCL script with

```
quartus_sh -t qvhdl.tcl > qvhdl.txt
```

The script produces for each design four parameters. For the `trisc0.vhd`, for instance, we get:

```
....
-----------------------------------------------------
trisc0 fmax: 115.65 MHz ( period = 8.647 ns )
trisc0 LEs: 198 / 33,216 ( < 1 % )
trisc0 M4K bits: 5,120 / 483,840 ( 1 % )
trisc0 DSP blocks: 1 / 70 ( 1 % )
-----------------------------------------------------
....
```

then `grep` through the report `qvhdl.txt` file using `fmax:`, `LEs:` etc.

From the script you will notice that the following special options of Quartus II web edition 6.0 were used:

- `Device` set `Family` to `Cyclone II` and then under `Available devices` select `EP2C35F672C6`.
- For `Timing Analysis Settings` set `Default required fmax:` to 3 ns.
- For `Analysis & Synthesis Settings` from the `Assignments` menu
  – set `Optimization Technique` to `Speed`
  – Deselect `Power-Up Don't Care`
- In the `Fitter Settings` select as `Fitter effort` `Standard Fit (highest effort)`

The table below displays the results for all VHDL and Verilog examples given in this book. The table is structured as follows. The first column shows the entity or module name of the design. Columns 2 to 6 are data for the VHDL designs: the number of LEs shown in the report file; the number of 9×9-bit multipliers; the number of M4K memory blocks; the `Registered Performance`; and the page with the source code. The same data are provided for the Verilog design examples, shown in columns 7 to 9. Note that VHDL and Verilog produce the same data for number of 9×9-bit multiplier and number of M4K memory blocks, but the LEs and `Registered Performance` do not always match.

			VHDL				Verilog	
Design	LEs	9 × 9	M4Ks	f_{MAX}	Page	LEs	f_{MAX}	Page
		Mult.		MHz			MHz	
add_1p	125	no	0	316.46	78	77	390.63	666
add_2p	234	no	0	229.04	78	144	283.85	667
add_3p	372	no	0	215.84	78	229	270.42	668
ammod	316	no	0	215.98	455	277	288.85	717
arctan	100	4	0	32.09	134	99	32.45	676
bfproc	131	3	0	95.73	370	83	116.09	715
ccmul	39	3	0	–	368	39	–	713
cic3r32	337	no	0	282.17	262	337	269.69	694
cic3s32	205	no	0	284.58	269	205	284.50	696
cmoms	372	10	0	85.94	303	239	107.48	706
cmul7p8	48	no	0	-	59	48	–	665
cordic	235	no	0	222.67	127	197	317.16	674
dafsm	32	no	0	420.17	189	30	420.17	683
dapara	33	no	0	214.96	202	45	420.17	691
darom	27	no	1	218.29	196	27	218.96	687
dasign	56	no	0	236.91	199	47	328.19	689
db4latti	418	no	0	58.81	324	248	74.69	709
db4poly	173	no	0	136.65	250	158	136.31	697
div_aegp	64	4	0	134.63	94	64	134.63	671
div_res	127	no	0	265.32	100	115	257.86	673
example	24	no	0	420.17	15	24	420.17	663
farrow	279	6	0	43.91	292	175	65.77	703
fir6dlms	138	4	0	176.15	511	138	174.52	721
fir_gen	184	4	0	329.06	167	184	329.06	681
fir_lms	50	4	0	74.59	504	50	74.03	719
fir_srg	114	no	0	97.21	179	70	106.15	682
fun_text	32	no	1	264.20	30	32	264.20	664
iir	62	no	0	160.69	217	30	234.85	692
iir_par	268	no	0	168.12	237	199	136.87	693
iir_pipe	124	no	0	207.08	231	75	354.48	692
lfsr	6	no	0	420.17	437	6	420.17	716
lfsr6s3	6	no	0	420.17	440	6	420.17	717
ln	88	10	0	32.76	145	88	32.76	677
mul_ser	121	no	0	256.15	82	140	245.34	670
rader7	443	no	0	137.06	355	404	159.41	710
rc_sinc	448	19	0	61.93	285	416	81.47	699
reg_file	211	no	0	-	559	211	–	723
sqrt	336	2	0	82.16	150	317	82.73	678
tris0	198	1	2	115.65	606	166	71.94	724