Examples: Version 8.7

In this directory the source files are processed with m4, dpic -g, and TikZ PGF. This is a collection of diagrams the author has had occasion to produce using m4 circuit macros and others, and gpic or dpic. The source-file names are shown for reference. There may be other or better m4 or pic constructs for producing the same drawings in some cases.

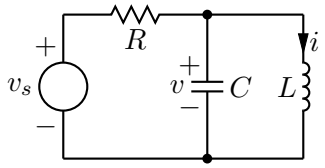


Figure 1: The quick-start example from the manual [quick.m4].

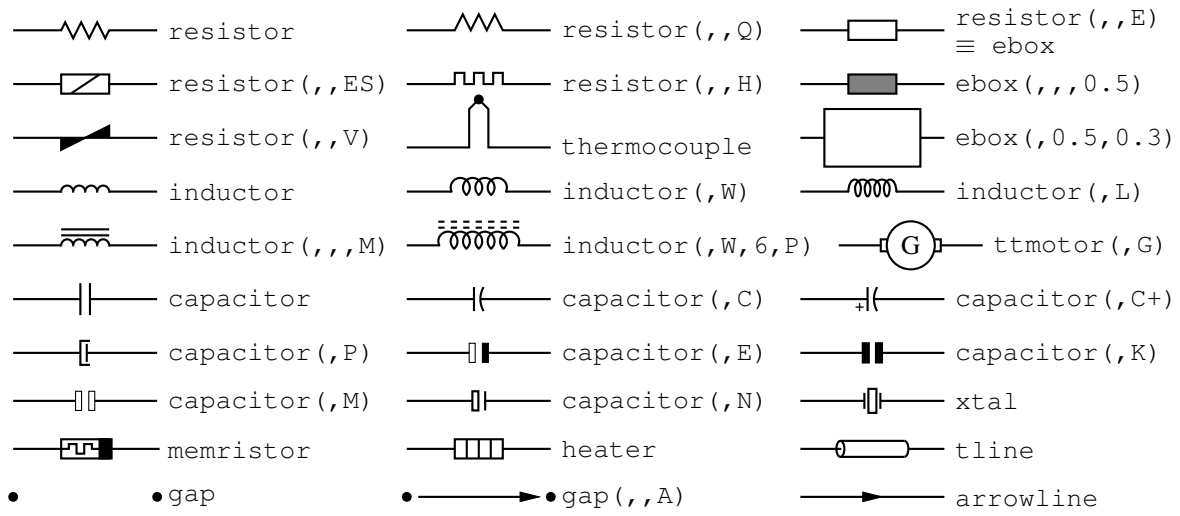


Figure 2: Two-terminal elements, showing some variations [CctTable.m4].

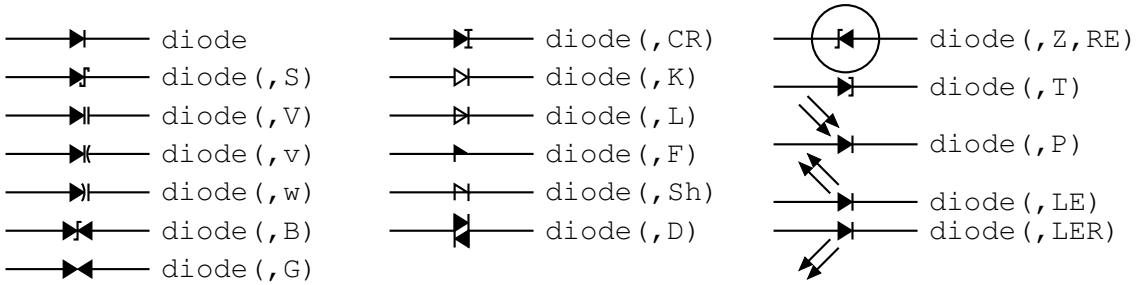


Figure 3: Diodes [Diodes.m4].

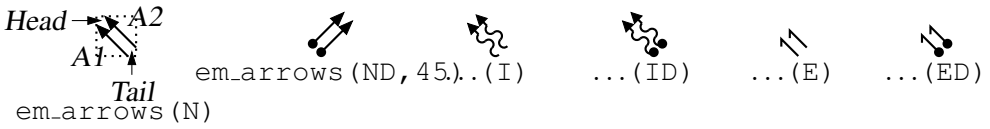


Figure 4: Radiation arrows [Emarrows.m4].

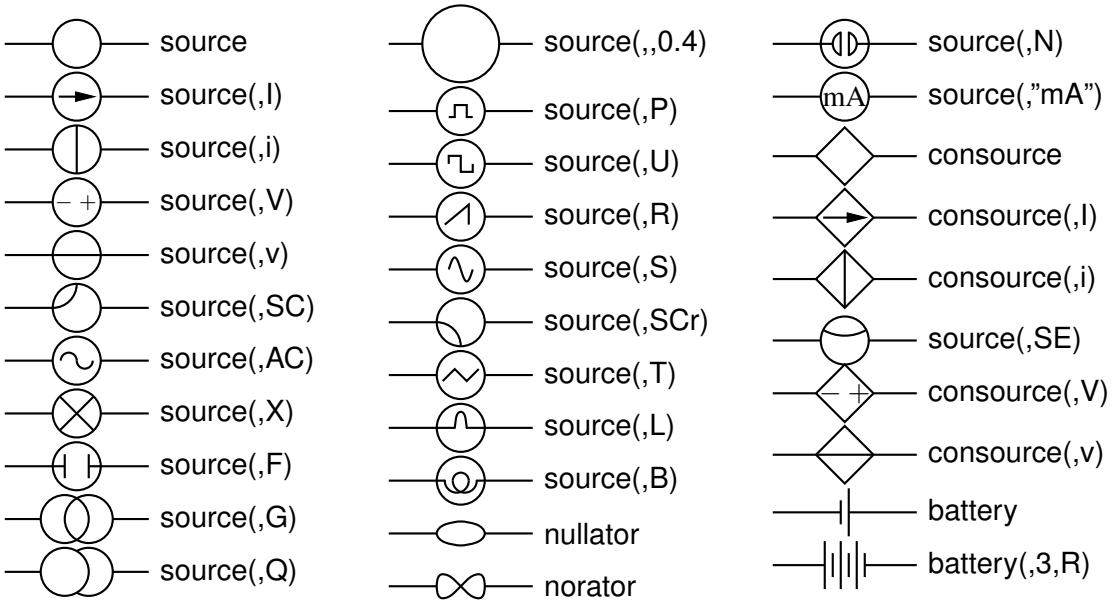


Figure 5: Sources and source-like elements `[Sources.m4]`.

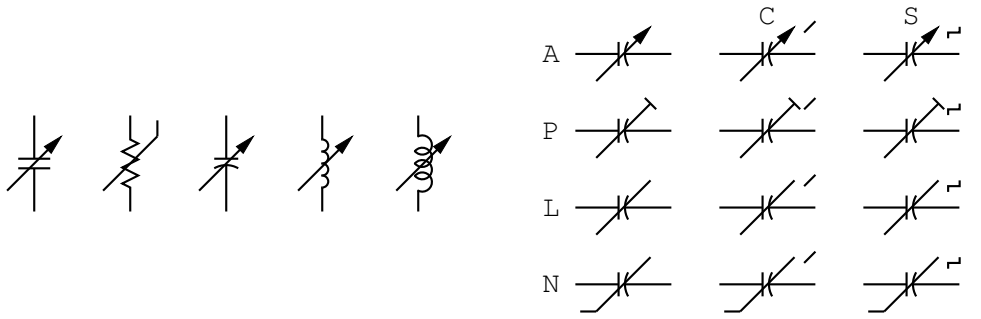


Figure 6: Arrows and marks indicating variability `[Variable.m4]`.

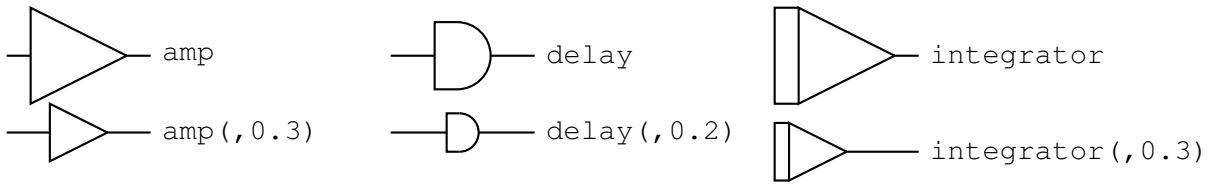


Figure 7: Macros `amp`, `delay`, and `integrator` `[AmpTable.m4]`.

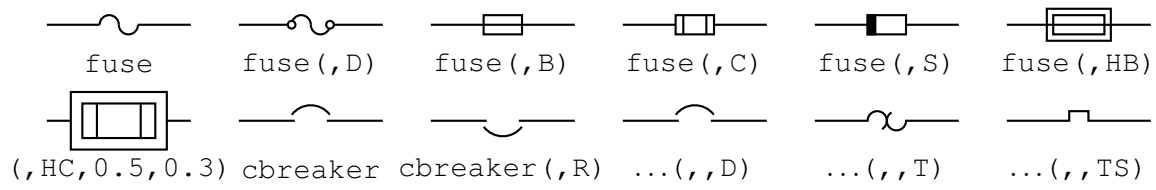


Figure 8: Macros `fuse` and `cbreaker` `[Fuses.m4]`.

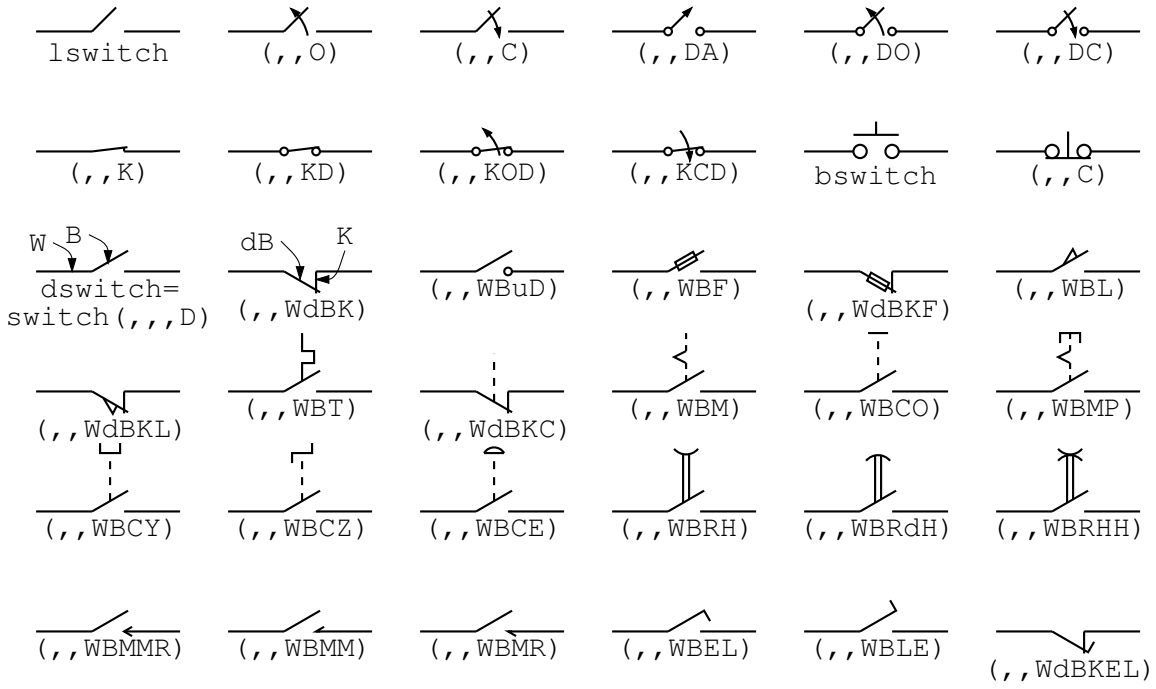


Figure 9: The switch macros; `switch(,,,L|B|D)` is a wrapper for `lswitch`, `bswitch`, and `dswitch` `[Switches.m4]`.

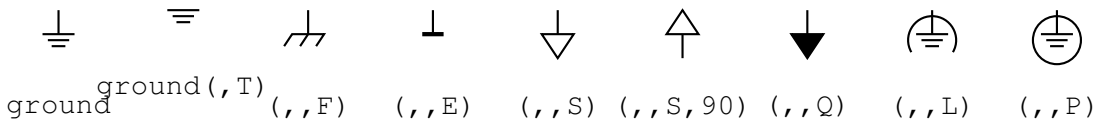


Figure 10: Ground symbols `[Grounds.m4]`.

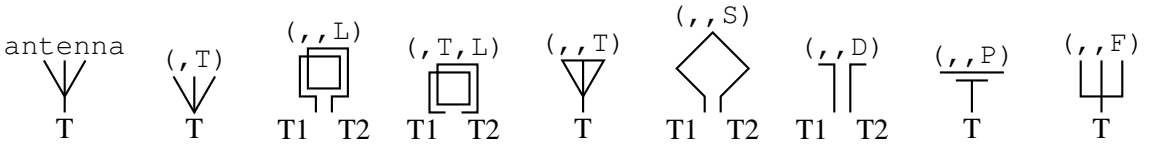


Figure 11: Antenna symbols `[Antennas.m4]`.

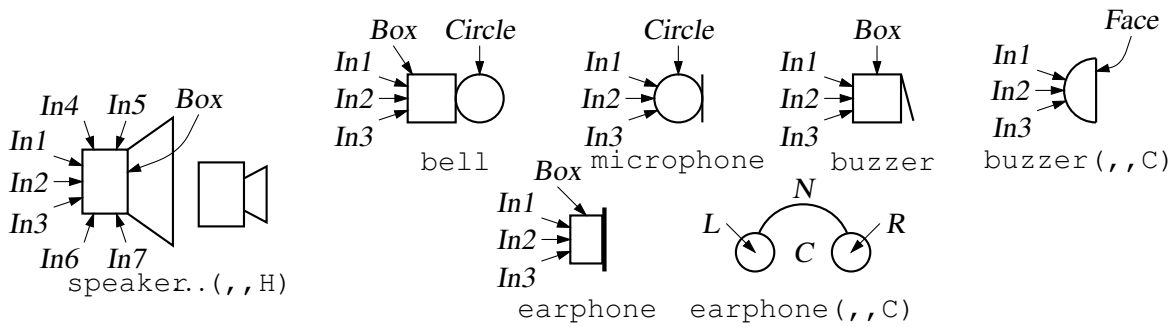


Figure 12: Audio elements `[Audio.m4]`.

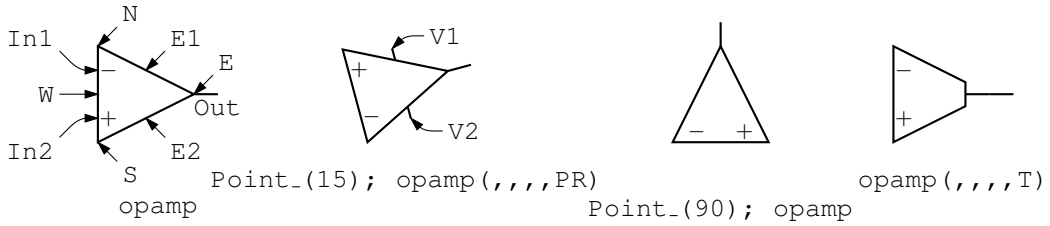


Figure 13: The opamp [Opamp.m4].

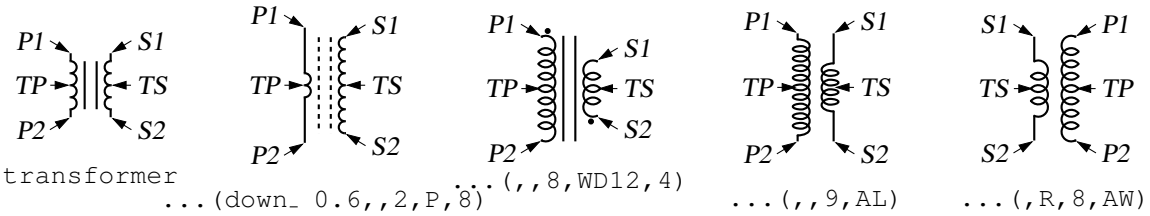


Figure 14: The transformer element, drawing direction down [Xform.m4].

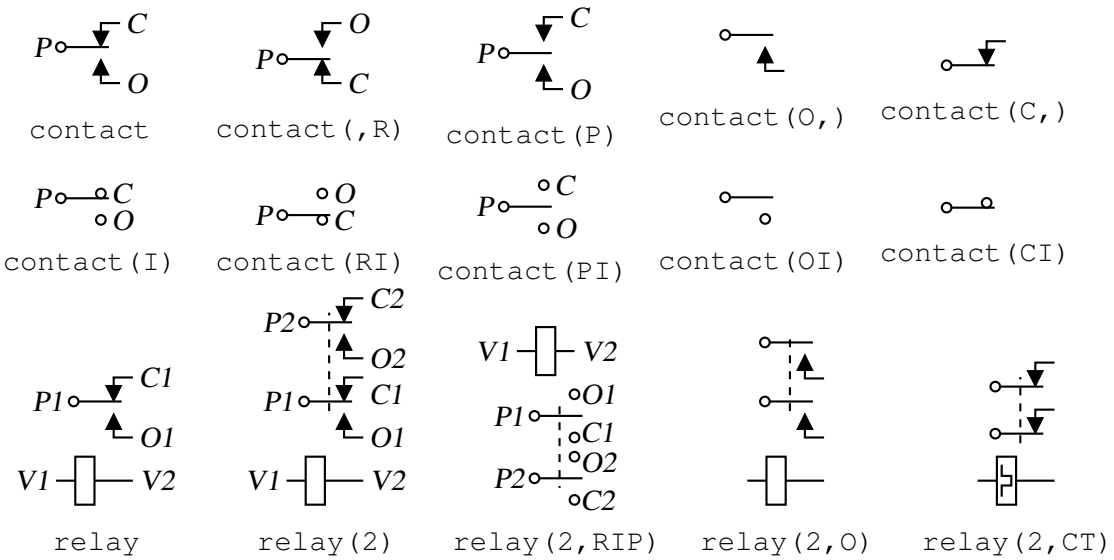


Figure 15: The contact and relay macros [Relay.m4].

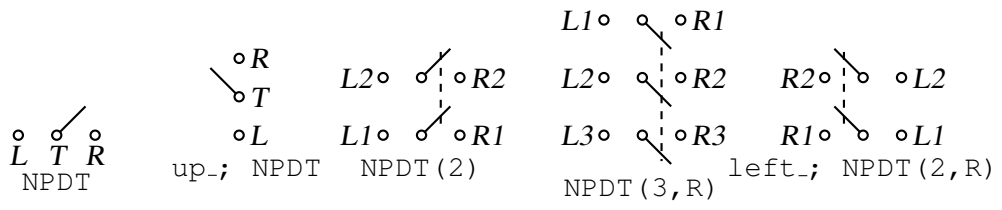


Figure 16: Double throw with the NPDT macro [NPDT.m4].

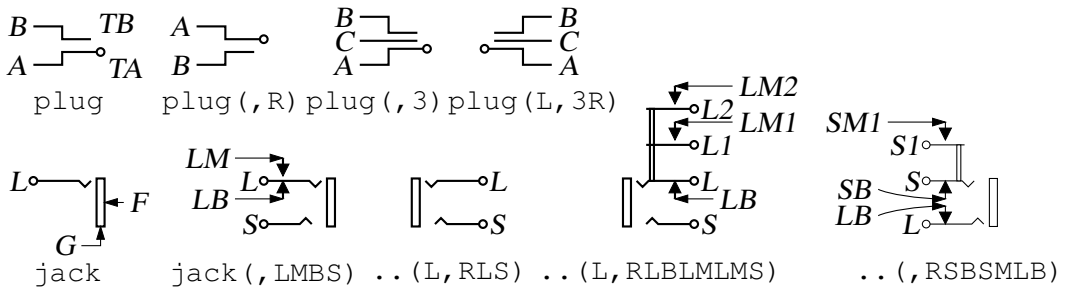


Figure 17: The `jack` and `plug` macros `[Jack.m4]`.

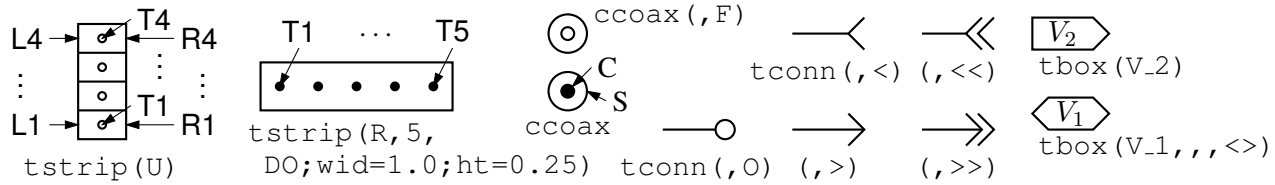


Figure 18: The `tstrip`, `ccoax`, `tconn`, and `tbox` macros `[Conn.m4]`.

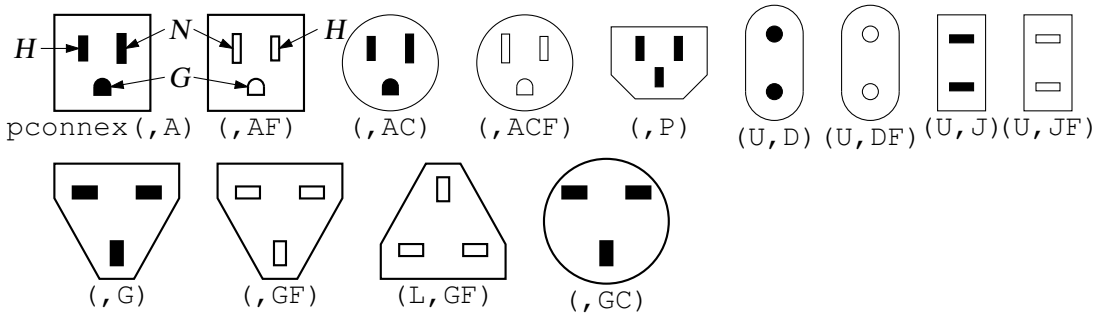


Figure 19: The `pconnex` macro `[Pconn.m4]`.

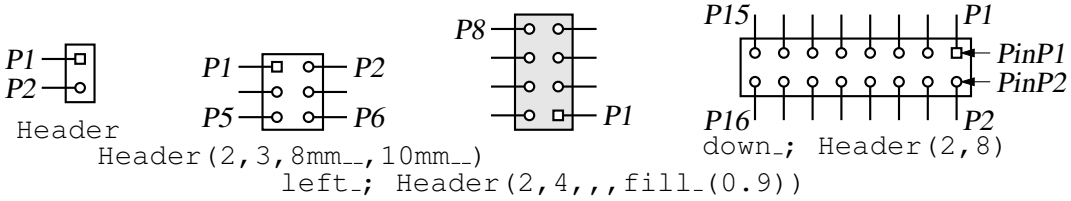


Figure 20: The `Header` macro `[Headers.m4]`.

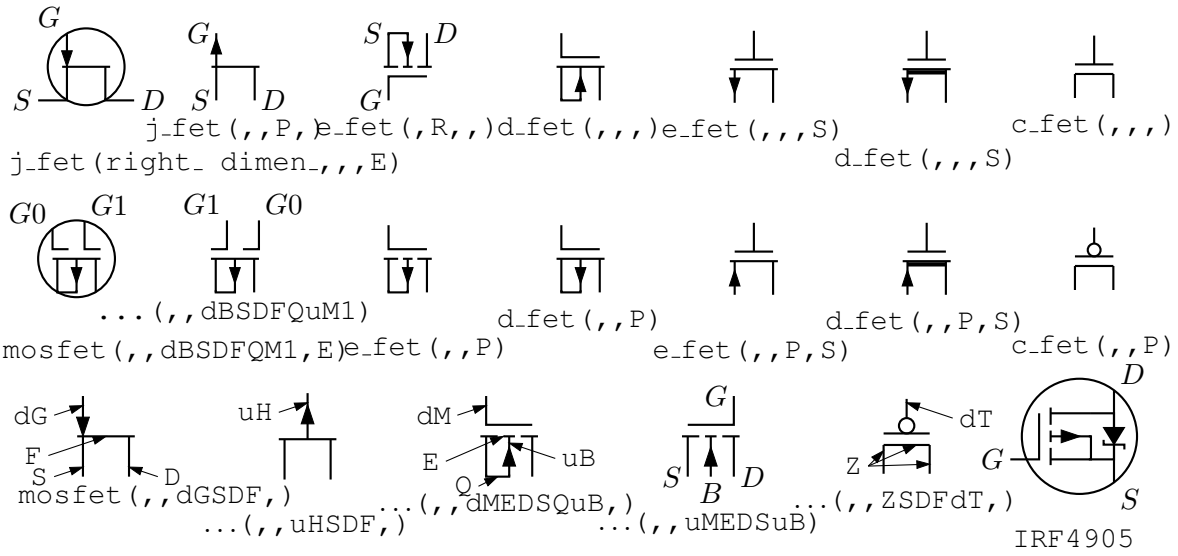


Figure 21: FETs, showing programmable components and example customizations [fet.m4].

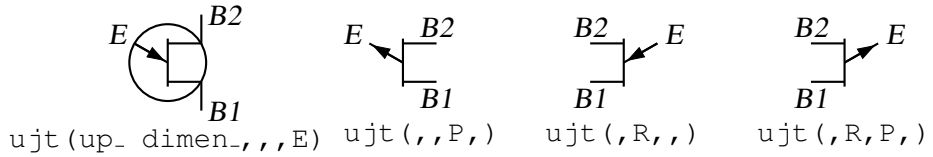


Figure 22: UJT examples [ujt.m4].

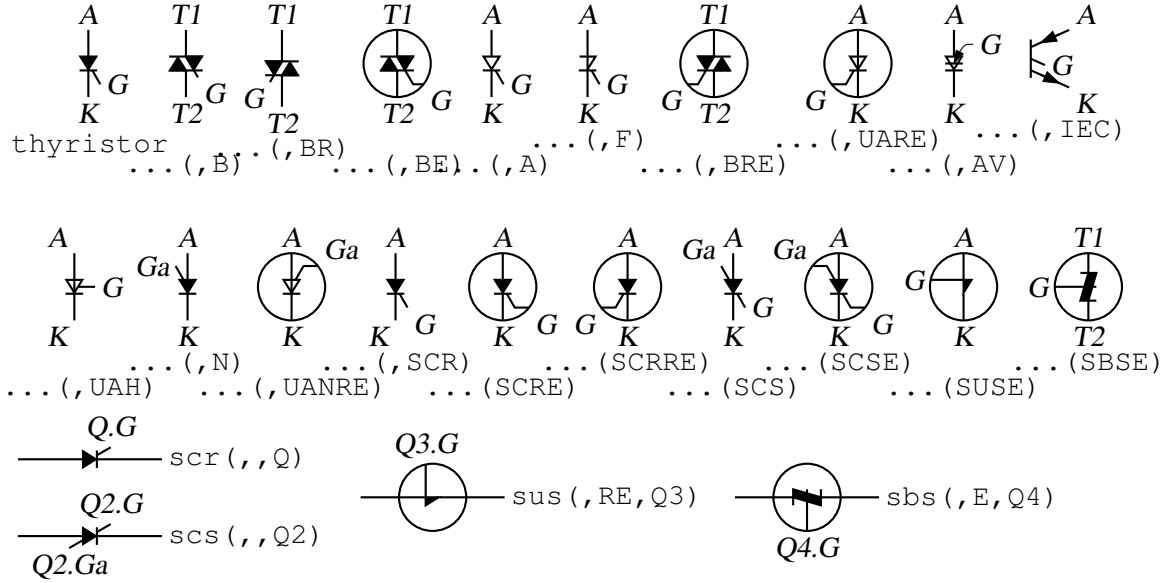


Figure 23: Thyristor examples [thyristor.m4].

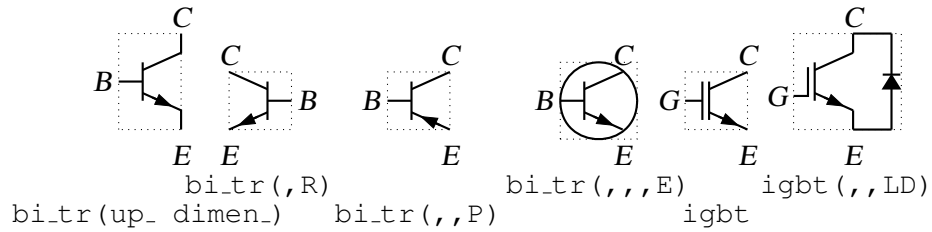


Figure 24: Bipolar transistors (drawing direction: up) [Bip.m4].

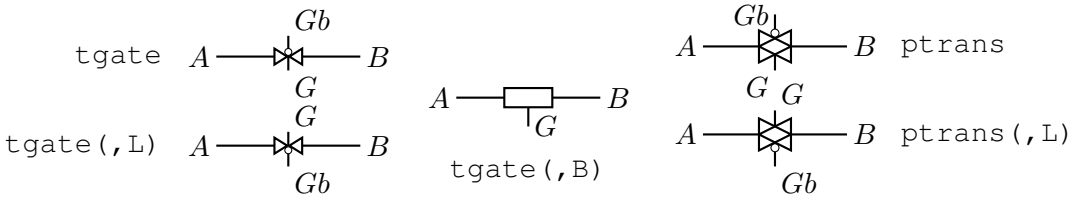


Figure 25: The tgate and ptrans elements [Tgate.m4].

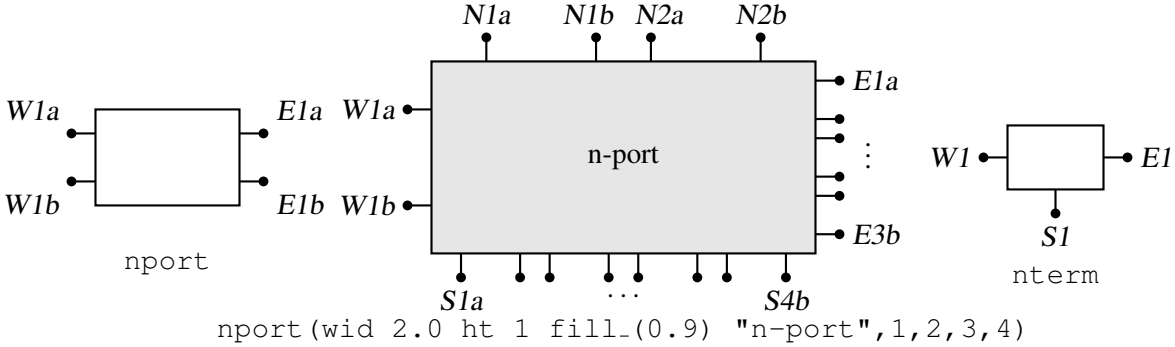


Figure 26: The nport and nterm macros [Nport.m4].

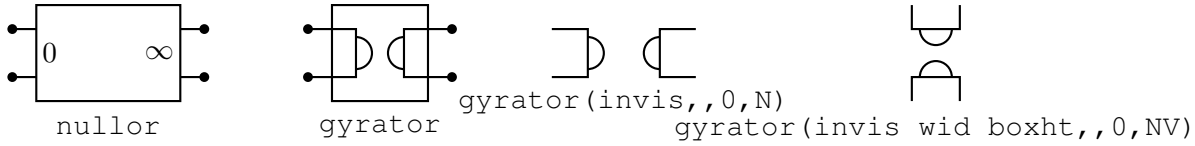


Figure 27: Some customizations of nport [NLG.m4].

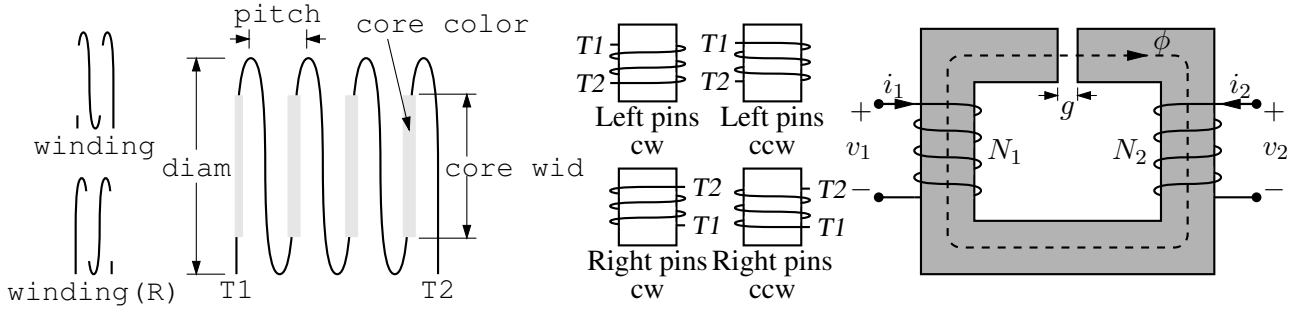


Figure 28: The macro winding(L|R,diam,pitch,turns,core wid,core color) [Windings.m4].

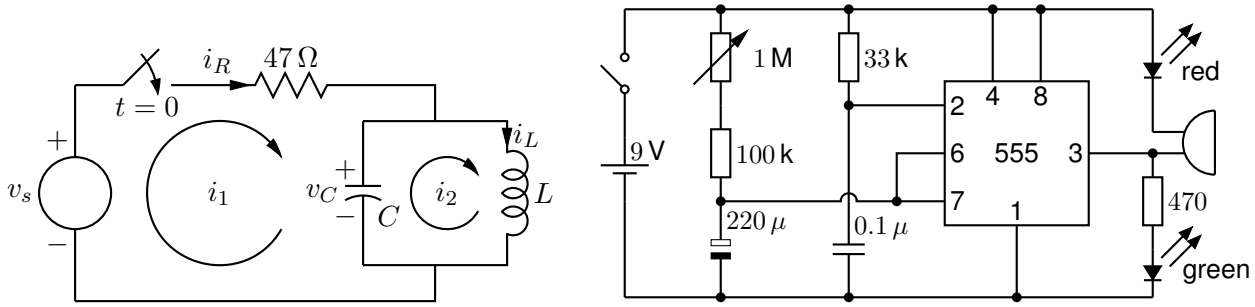


Figure 29: Two simple labeled circuits [ex01.m4][Timer.m4].

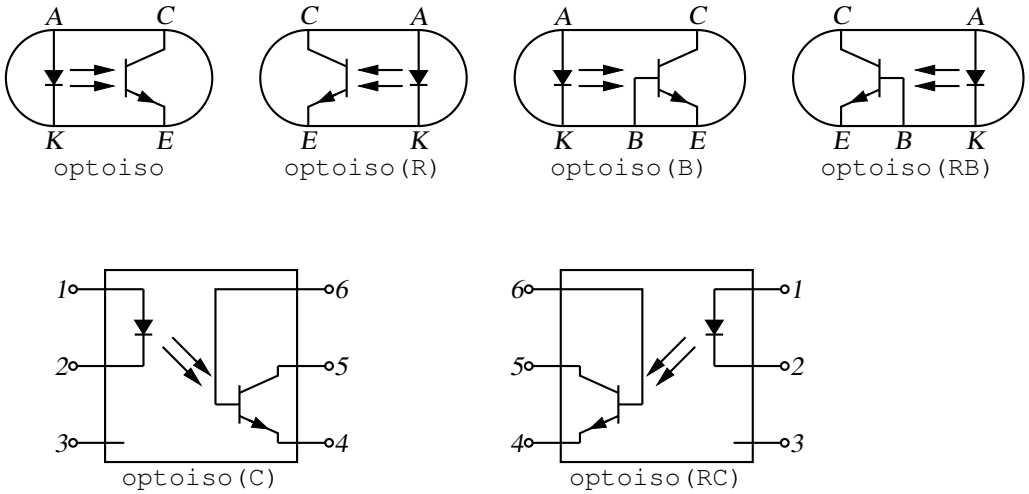


Figure 30: Optical isolator: a circuit with right or left orientation [Optoiso.m4].

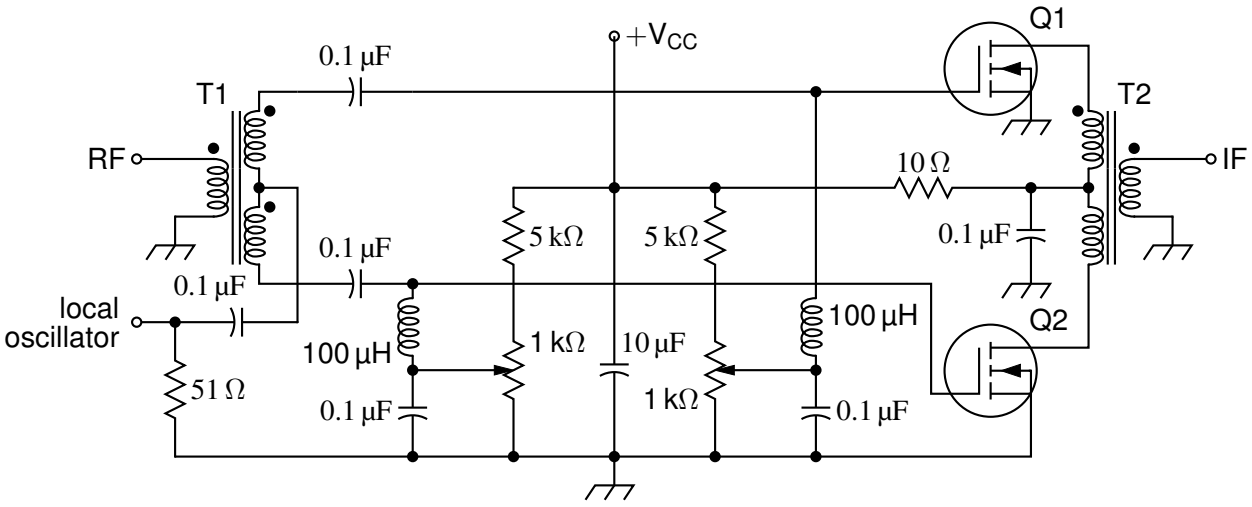


Figure 31: A balanced mixer, using mosfet and a custom transformer [Mixer.m4].

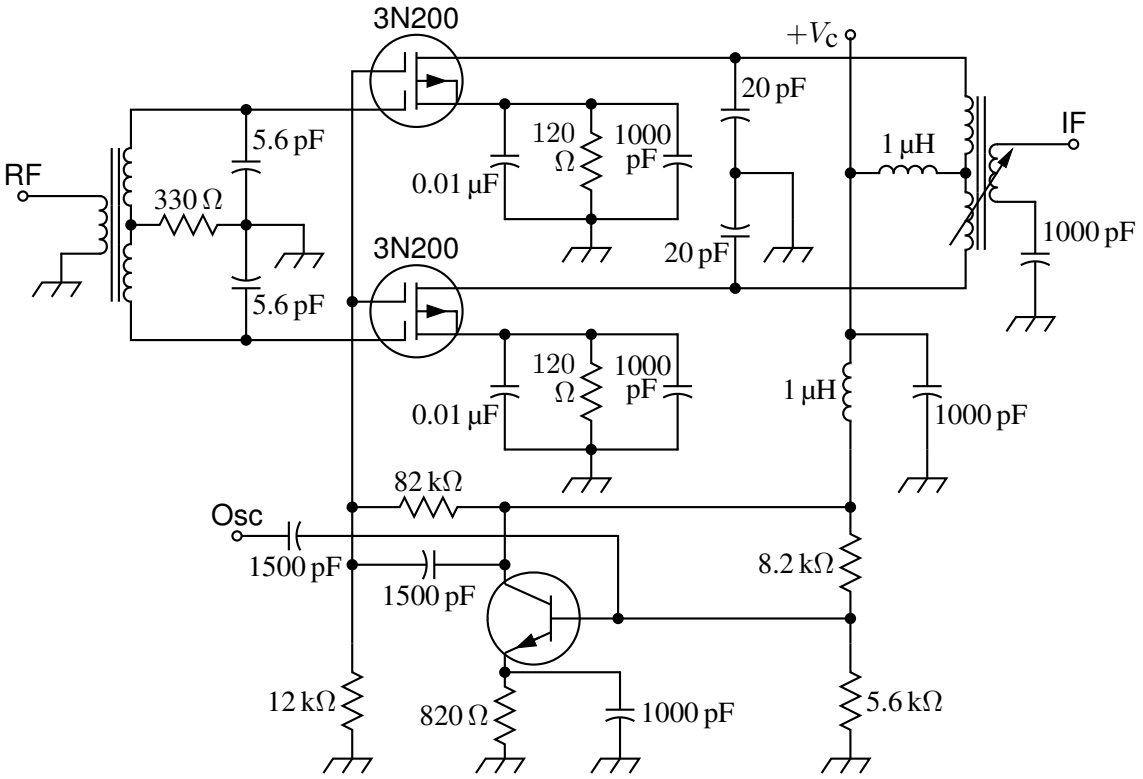


Figure 32: A push-pull mixer, showing FETs with multiple gates [PushPull.m4].

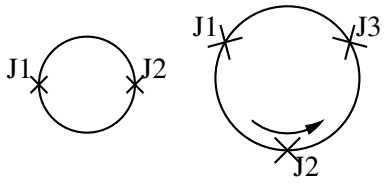


Figure 33: Superconducting quantum interface device (drawing direction down) [SQUID.m4].

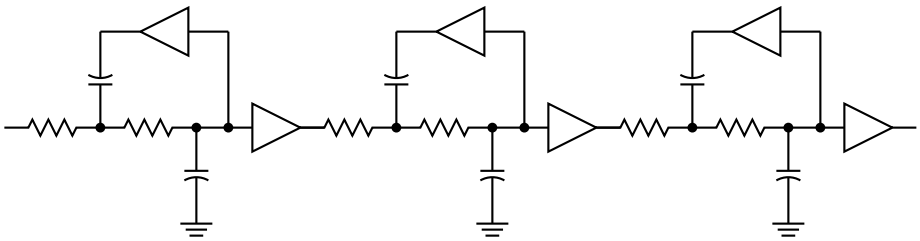


Figure 34: A six-pole filter [Sixpole.m4].

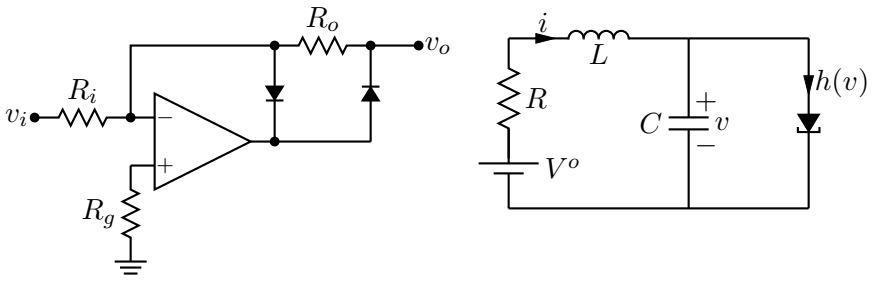


Figure 35: Precision half-wave rectifier and a tunnel diode circuit (illustrating `opamp`, `diode`, `resistor`, `ground`, and labels) `[ex18.m4]`.

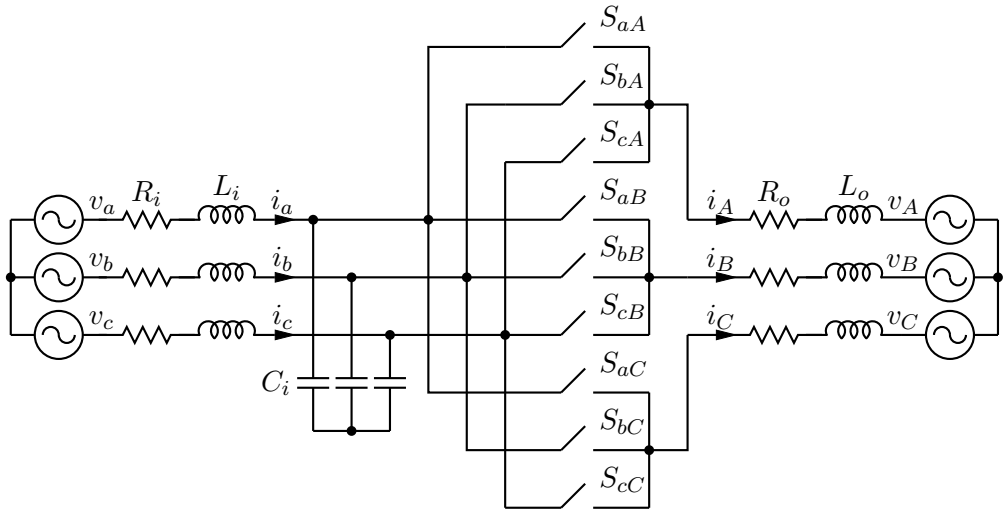


Figure 36: A three-phase switched AC-AC converter `[MC.m4]`.

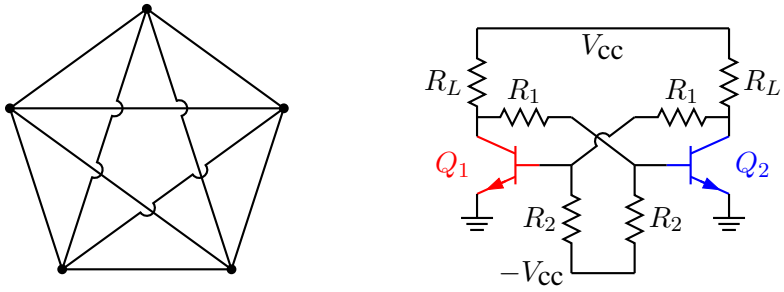


Figure 37: Non-planar graph and bistable circuit (illustrating the `crossover` macro and colored elements) `[ex10.m4]`.

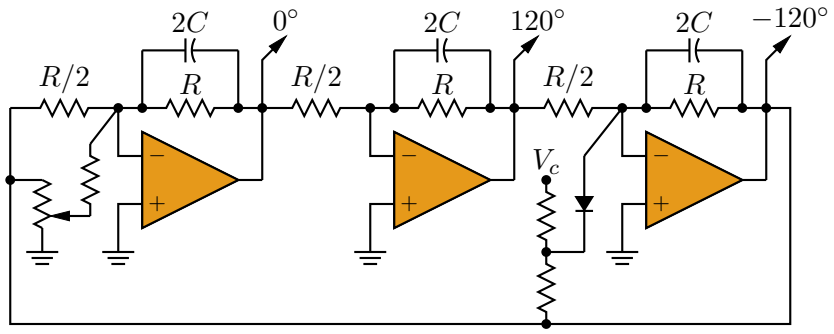


Figure 38: Three-phase oscillator [Three.m4].

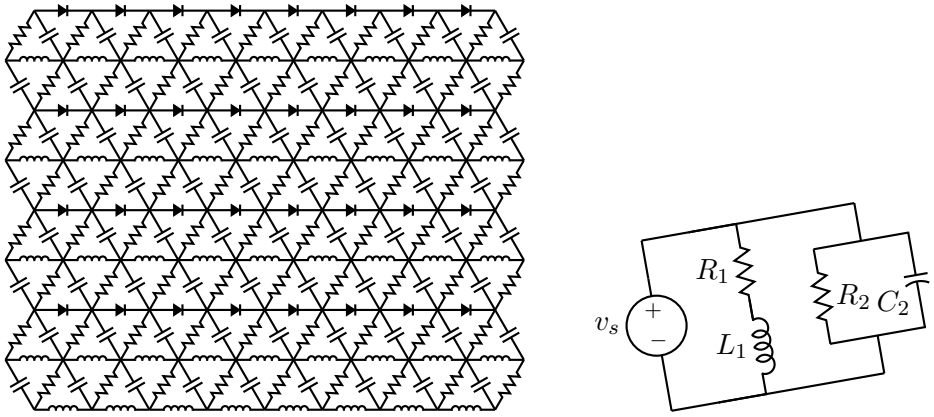


Figure 39: A repetitive network created by Pic looping and a skewed circuit used to test the macro parallel_ [ex17.m4].

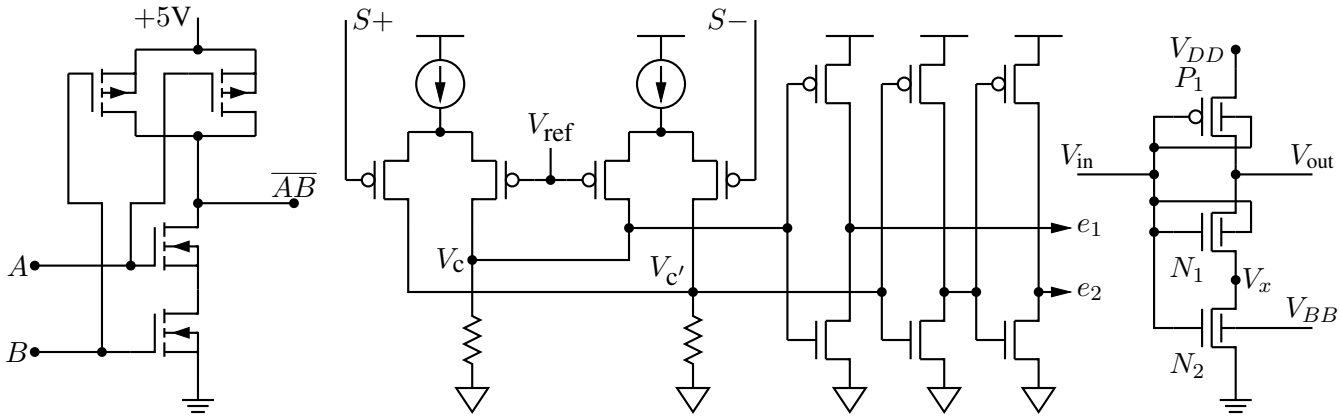


Figure 40: A CMOS NAND gate, a test circuit, and an XMOSFET example [ex12.m4].

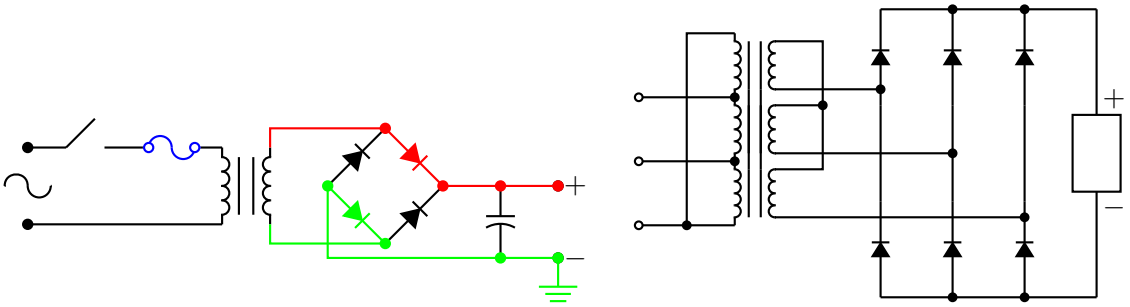


Figure 41: An elementary power supply circuit with colored elements, and a multiple-winding transformer with 3-phase rectifier `[pwrsupply.m4]`.

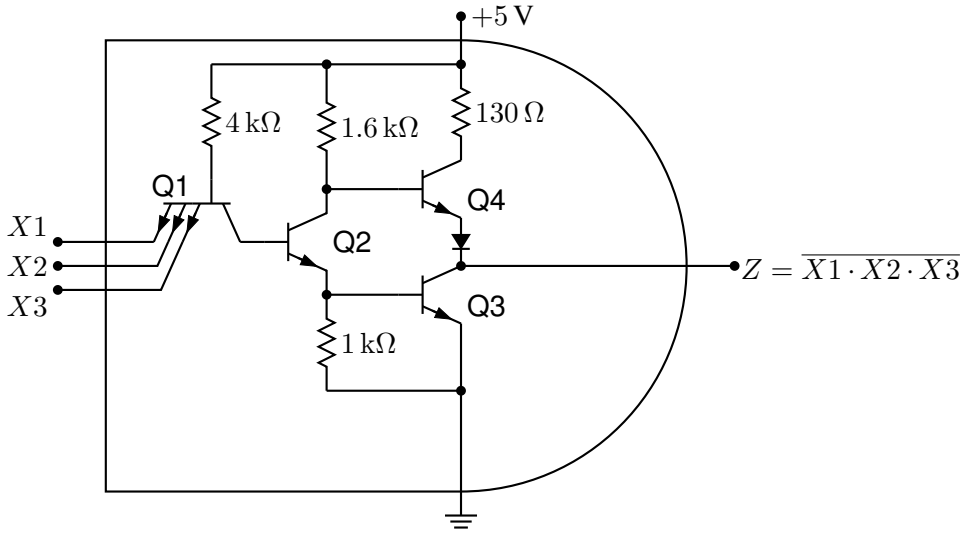


Figure 42: TTL NAND gate illustrating a transistor with multiple emitters `[TTLnand.m4]`.

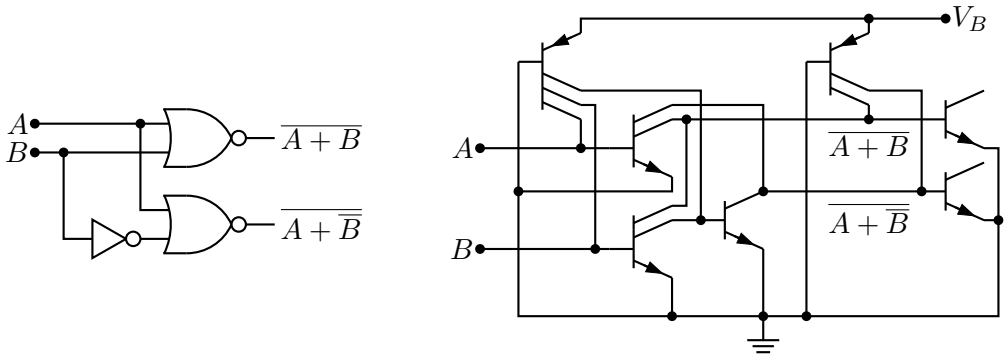


Figure 43: Gate circuit and equivalent embedded I^2L components illustrating multiple collectors `[I2L.m4]`.

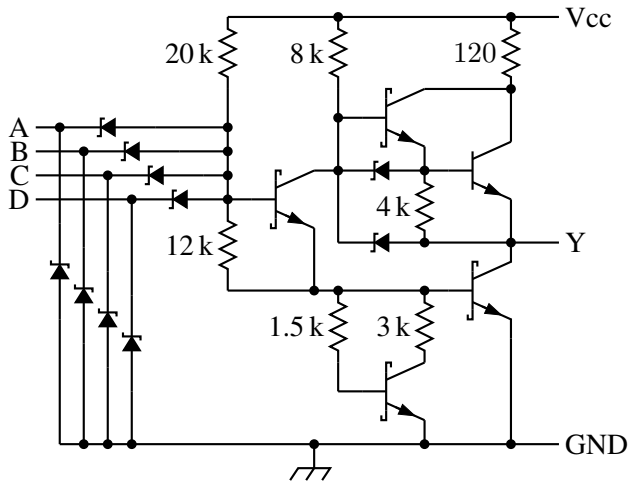


Figure 44: A 4-input NAND circuit illustrating the `S` (Schottky) option of `bi_trans [Schottky.m4]`.

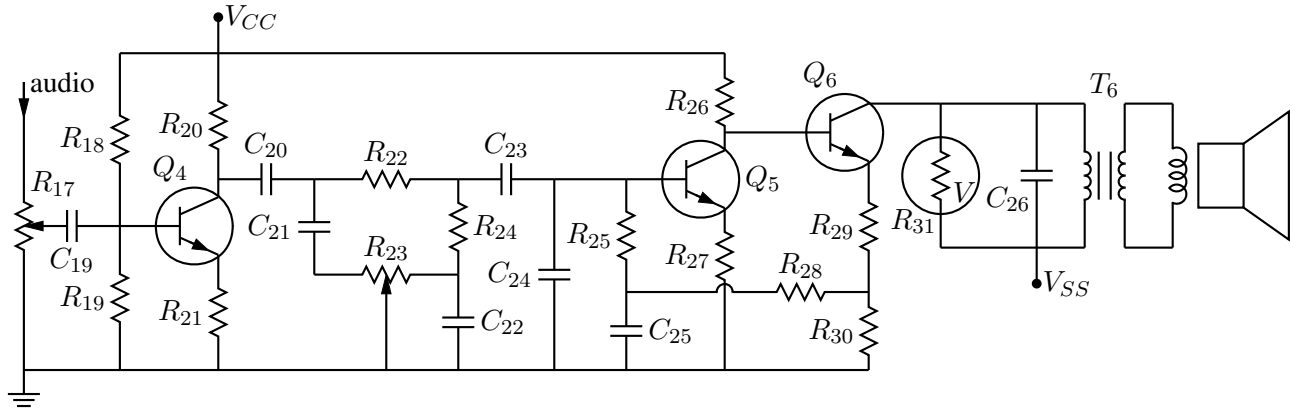


Figure 45: Transistor radio audio chain `[ex11.m4]`.

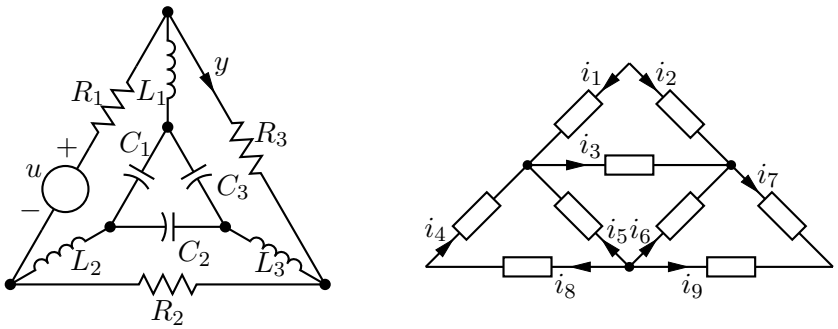


Figure 46: Labels on non-manhattan elements `[ex04.m4]`.

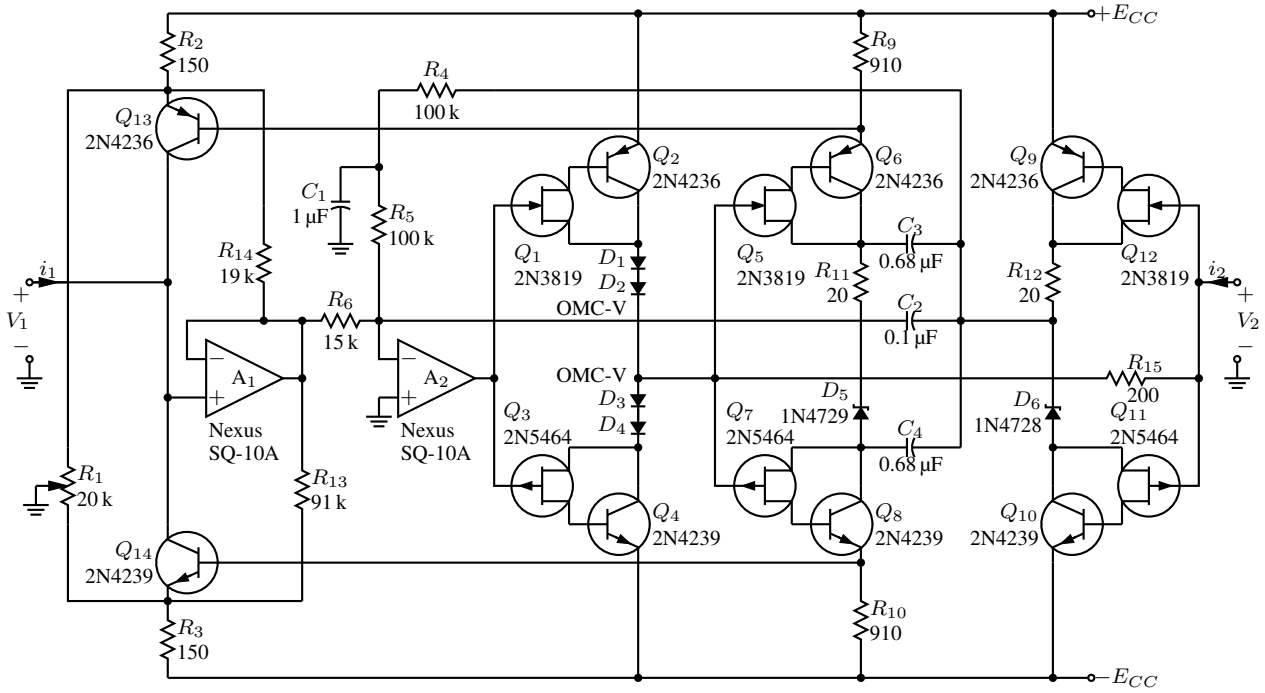


Figure 47: Realization of a controlled source (illustrating stacked element labels) [Csource.m4].

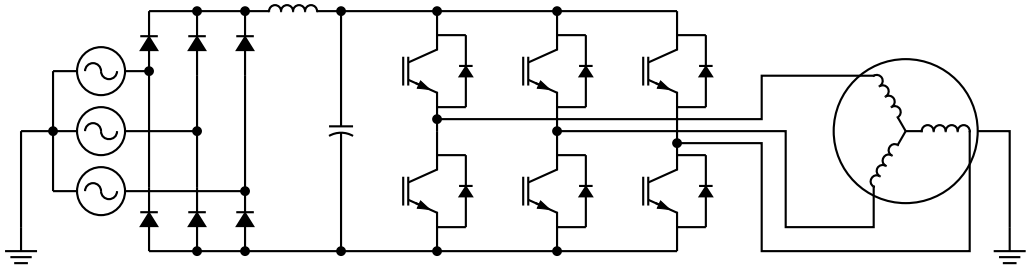


Figure 48: Synchronous machine driven by variable-speed drive and rectifier [Drive.m4].

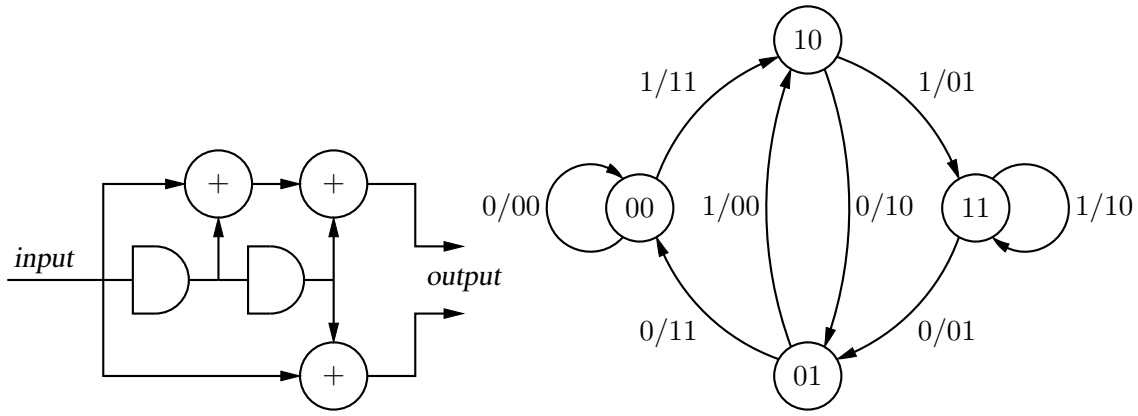


Figure 49: A rate $1/2$ binary convolutional coder and its state diagram [ex16.m4].

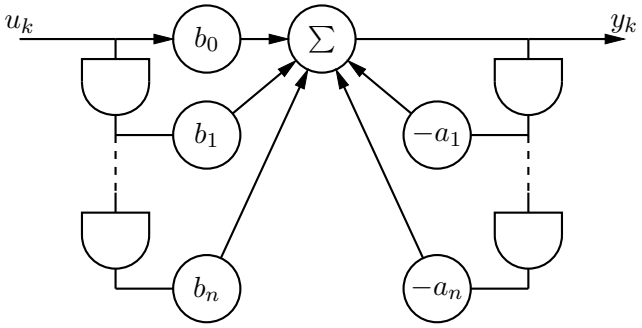


Figure 50: Digital filter [ex03.m4].

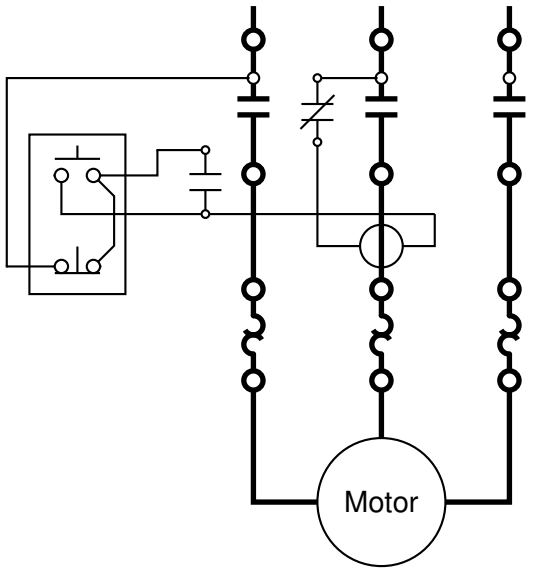


Figure 51: Motor control connections [MotorControl.m4].

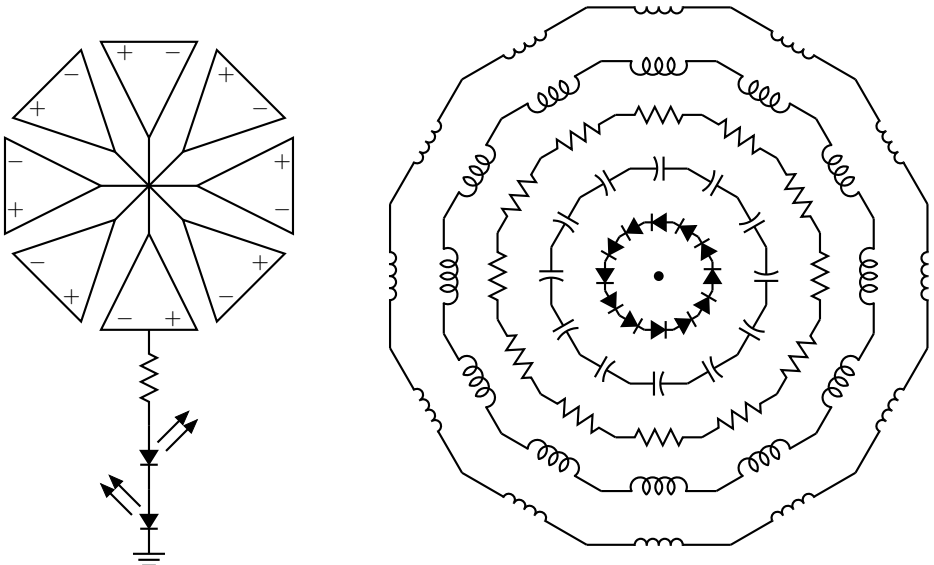


Figure 52: Elements at obtuse angles [ex02.m4].

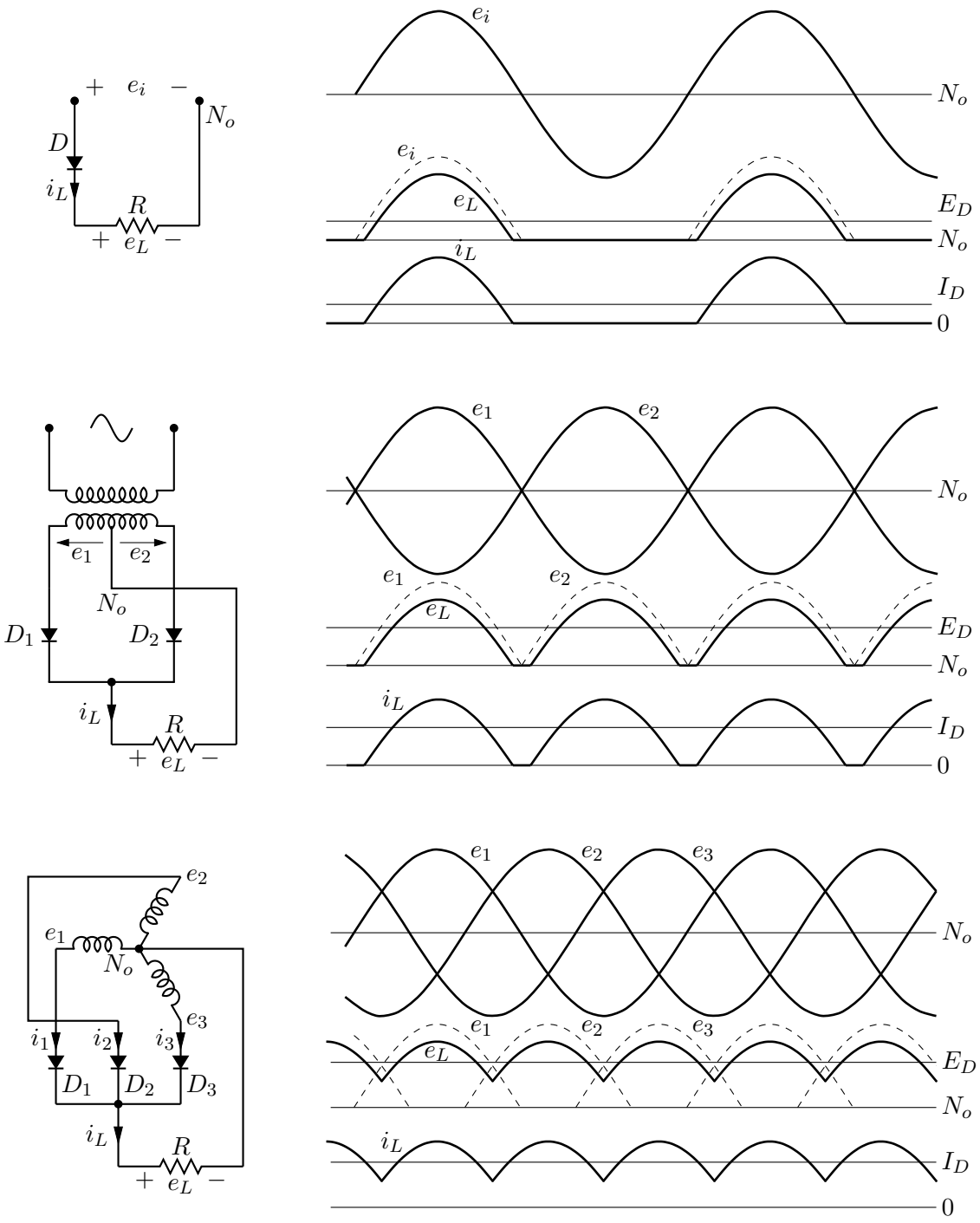


Figure 53: Rectifier circuits and waveforms `[Rectifiers.m4]`.

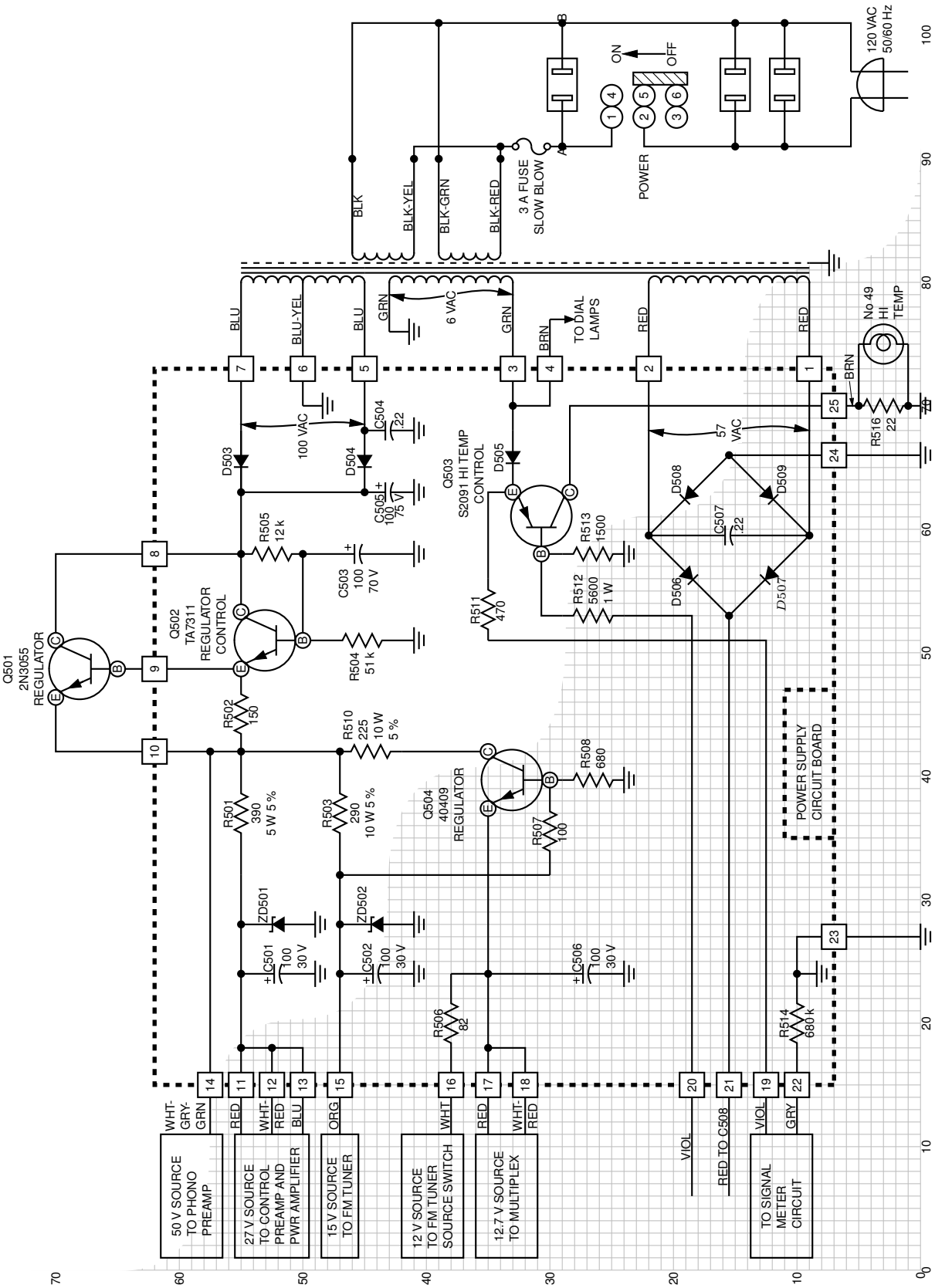


Figure 54: The power supply of a Heathkit AR-15 (Now, *that* was a receiver!) with custom transformer and other elements, drawn on a grid (partially shown) to aid in placement `[Heathkit.m4]`.

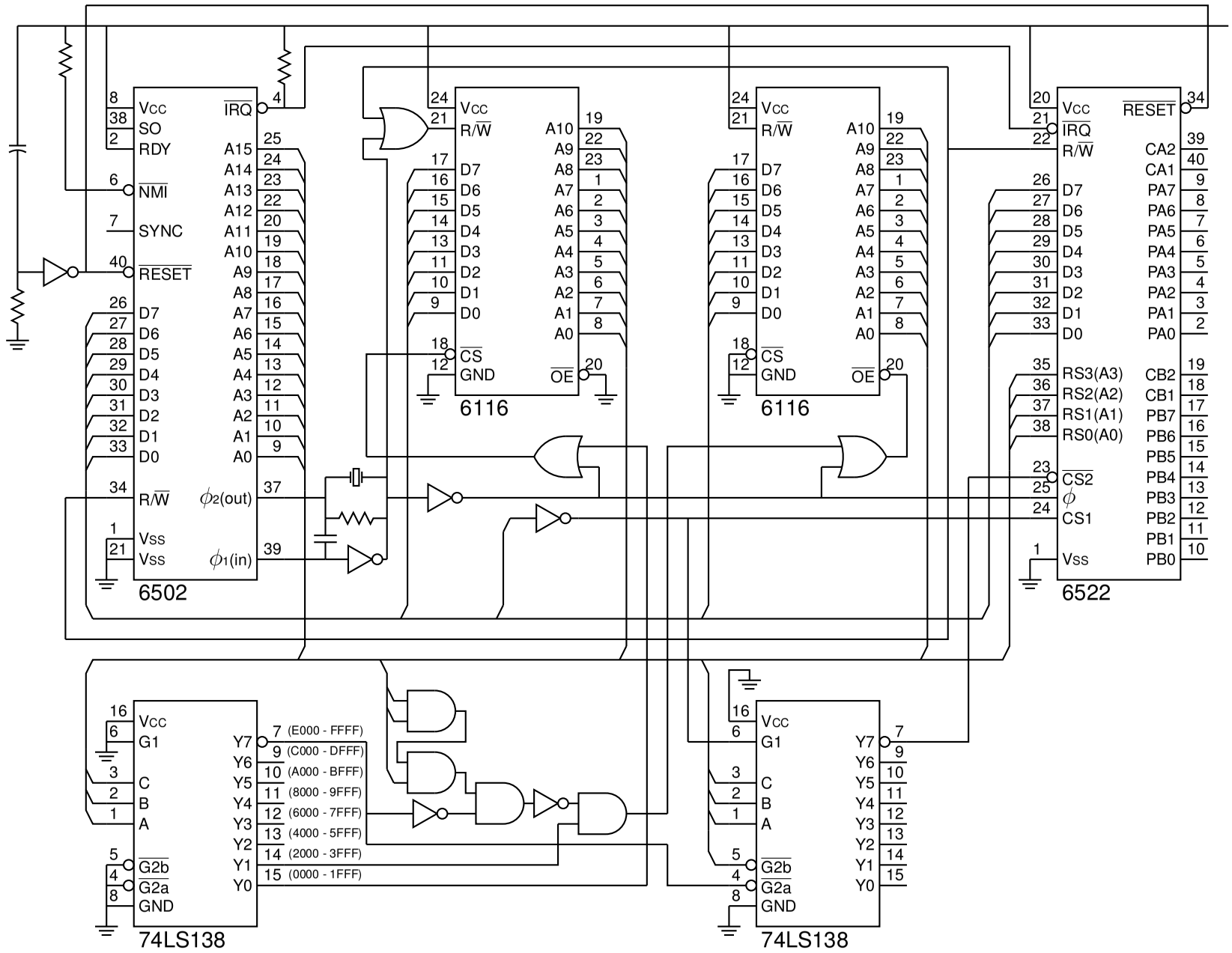


Figure 55: A digital circuit of moderate size, redrawn from M. P. Maclenan and G. M. Burns, "An Approach to Drawing Circuit Diagrams for Text Books," Tugboat (12)1, March 1991, pp. 66-69 `[lcct.m4]`.

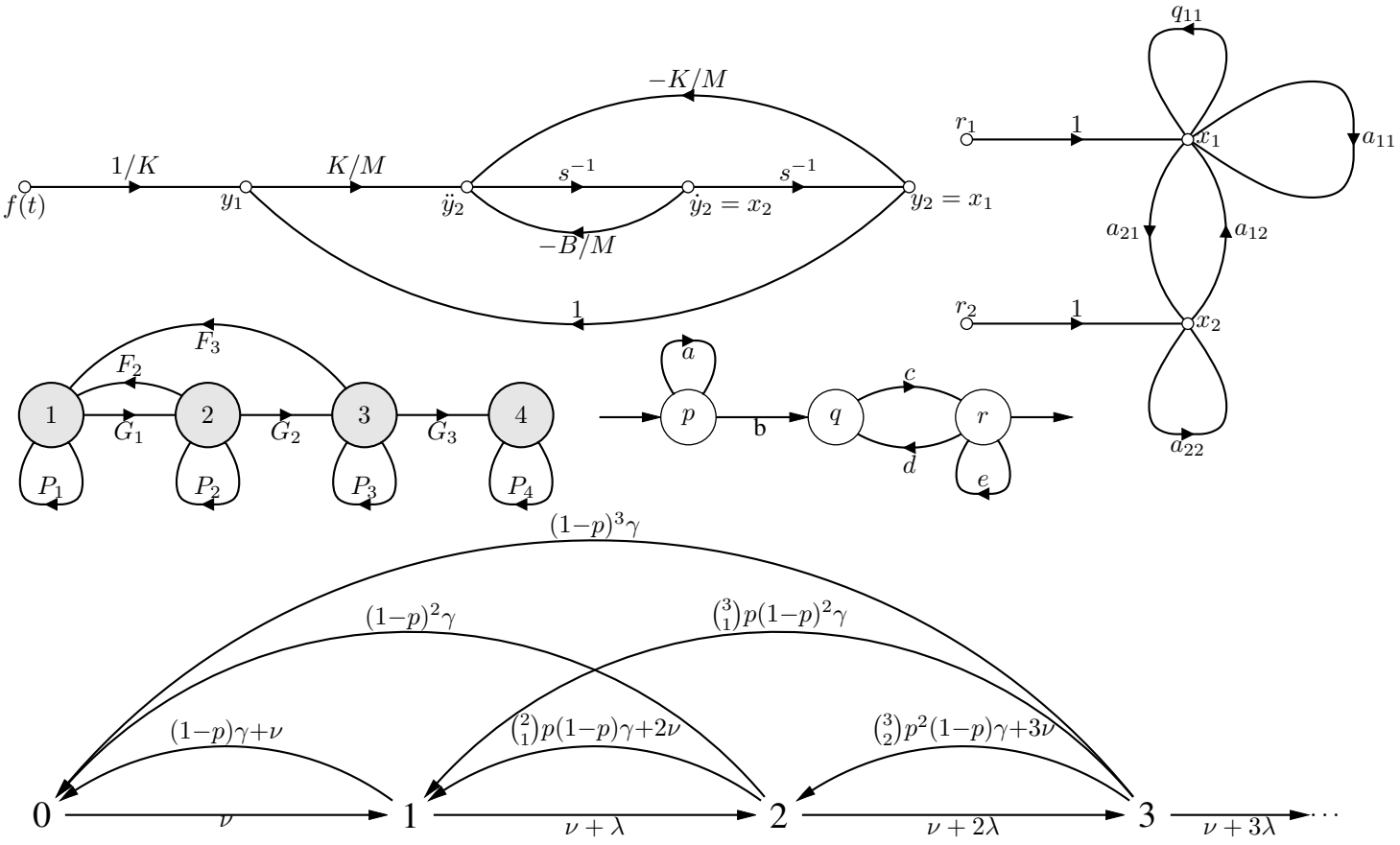


Figure 56: Signal-flow graphs `[sfg.m4]`.

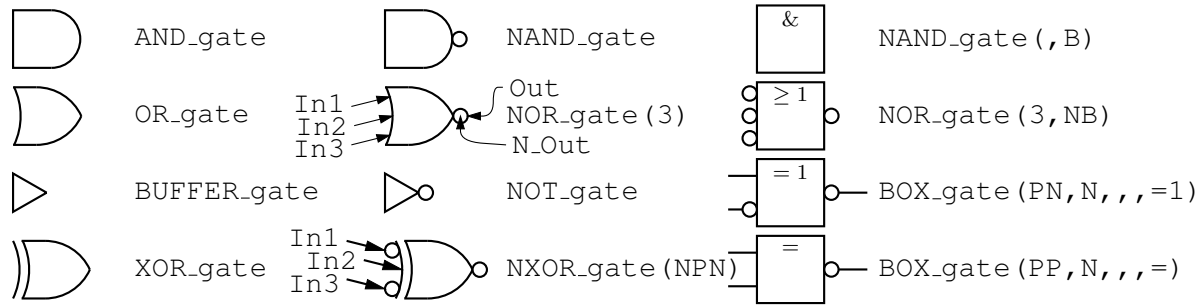


Figure 57: Basic logic gates `[Logic.m4]`.

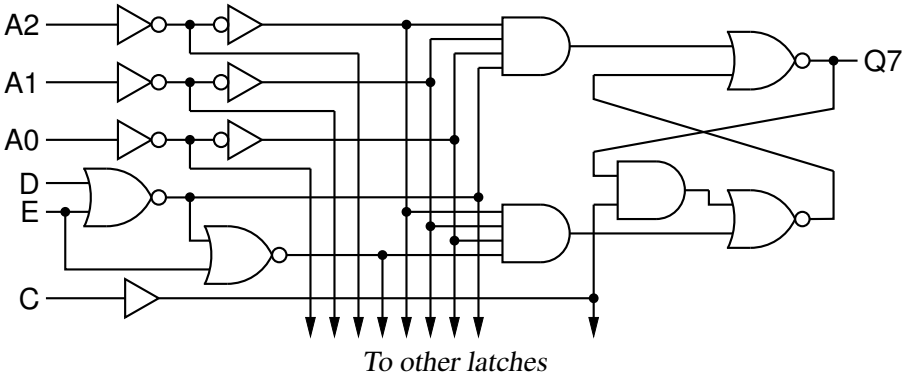


Figure 58: General-purpose latch: a small logic circuit [ex08.m4].

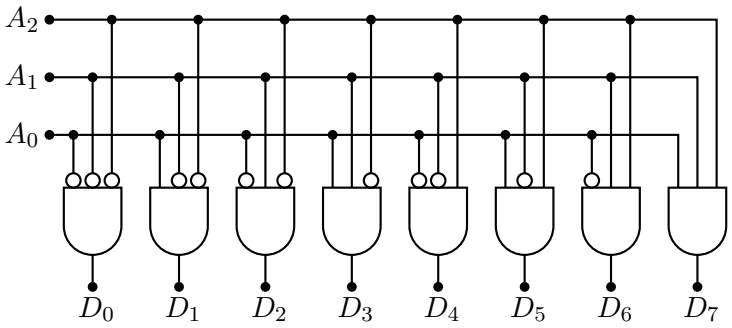


Figure 59: Decoder logic, constructed using the for_ macro [Decoder.m4].

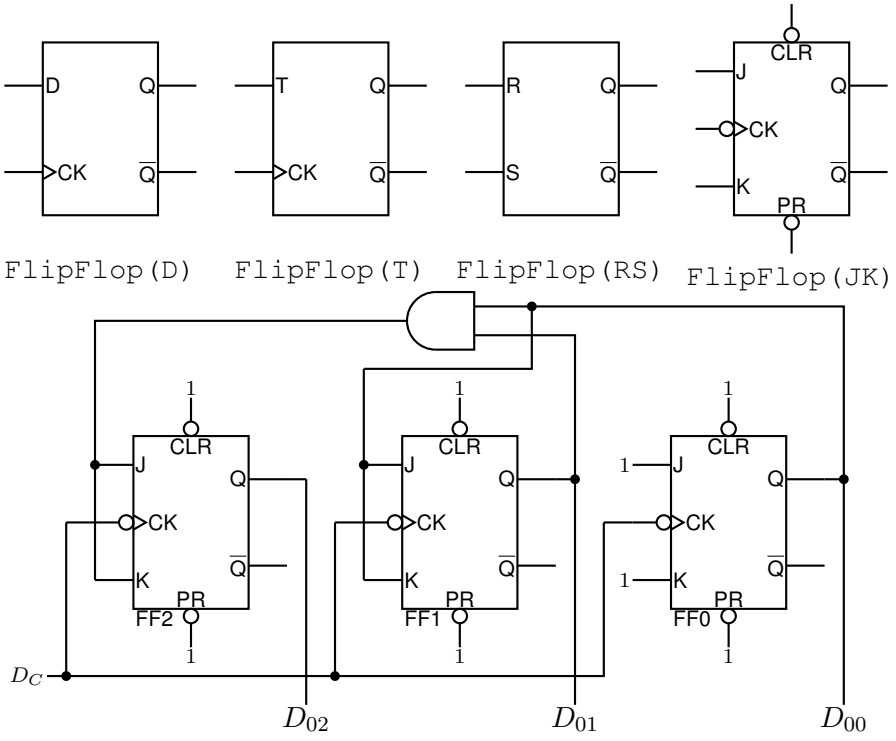


Figure 60: Some flip-flops [ex21.m4].

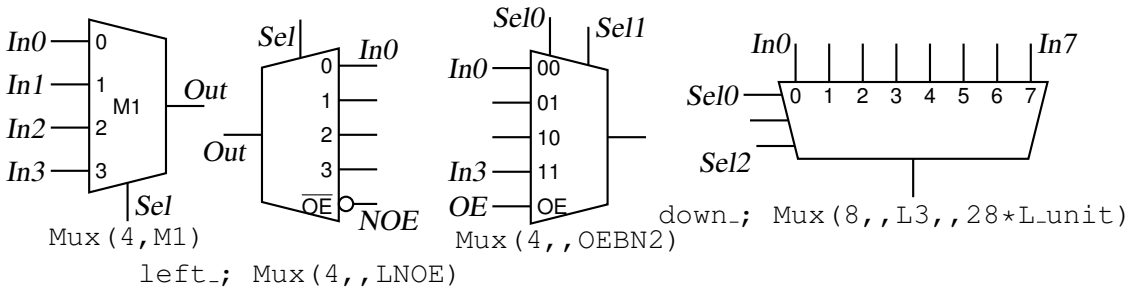


Figure 61: Multiplexer [Multiplexer.m4].

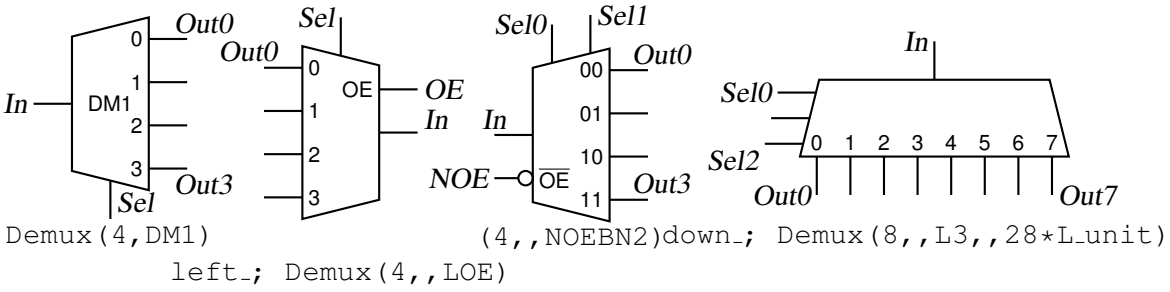


Figure 62: Demultiplexer [Demultiplexer.m4].

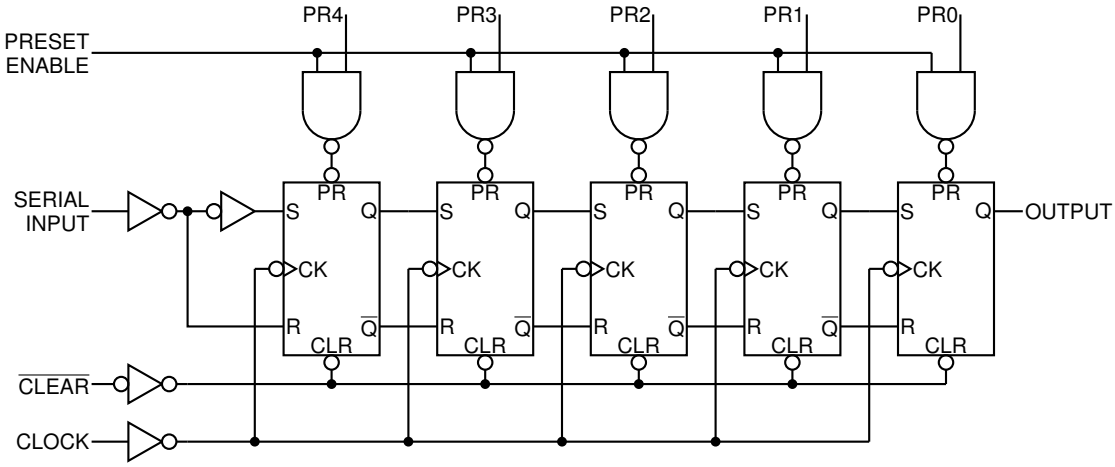


Figure 63: A 5-bit shift register drawn using a custom flip-flop [ShiftR.m4].

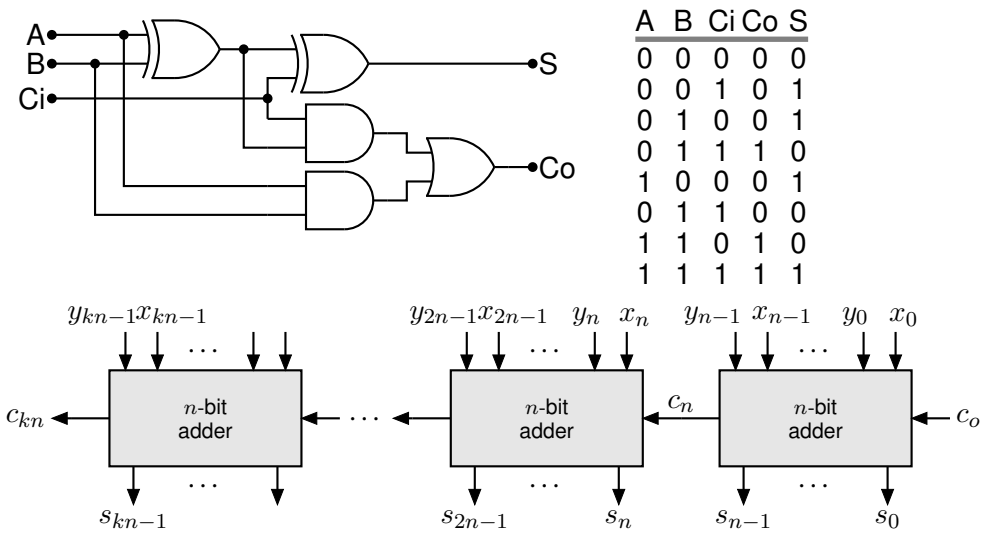
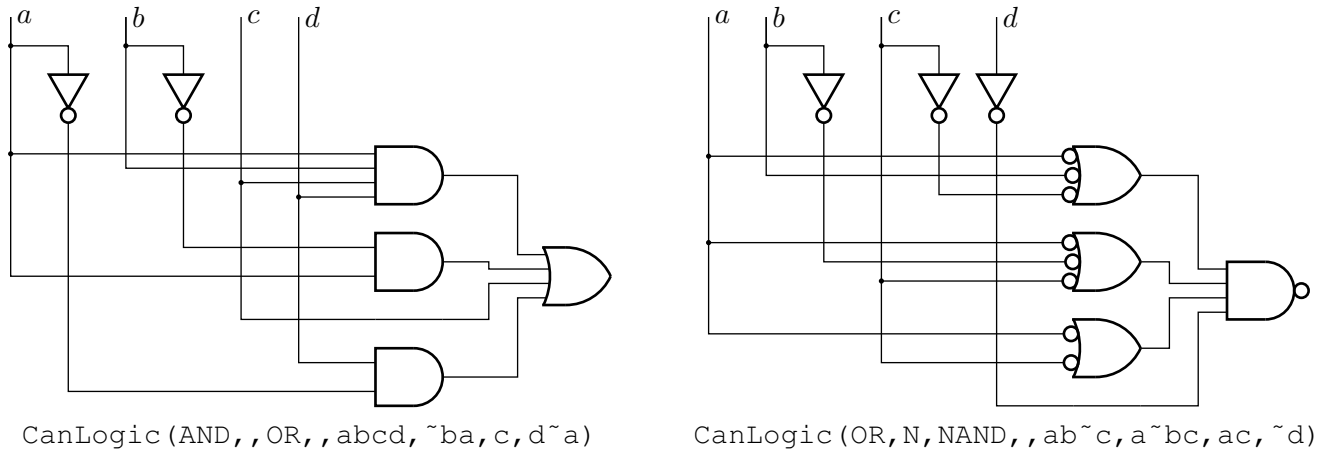


Figure 64: A full adder and a cascade of n-bit adders `[Adder.m4]`.



`CanLogic(AND,,OR,,abcd,~ba,c,d~a)` `CanLogic(OR,N,NAND,,ab~c,a~bc,ac,~d)`

Figure 65: A way of automatically drawing two-layer logic diagrams `[CanLogic.m4]`.

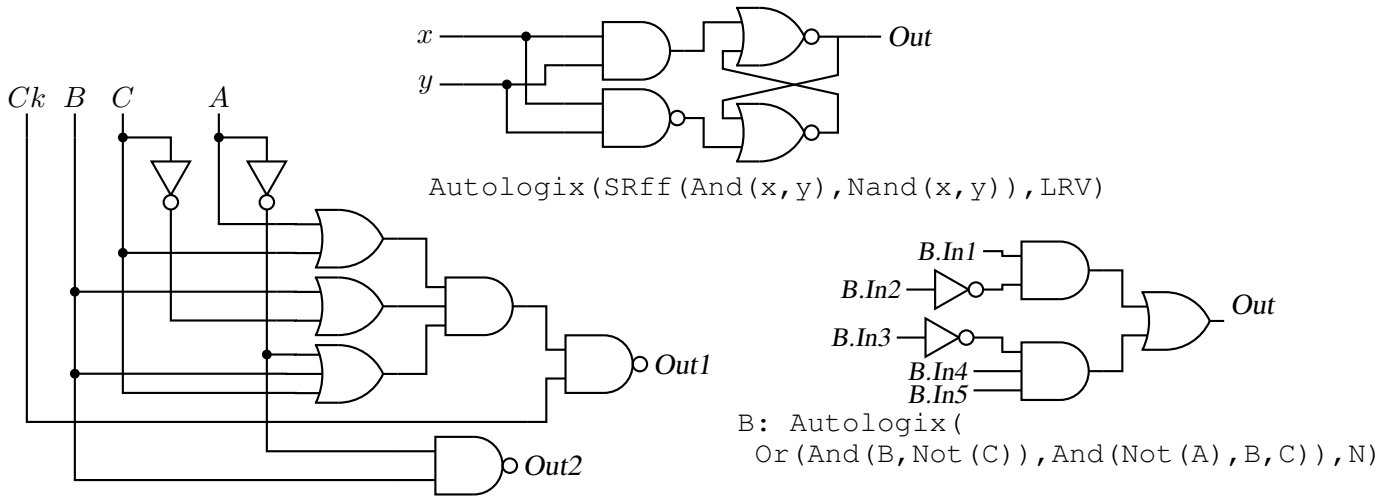


Figure 66: The `Autologix(`*Boolean expression*; *Boolean expression...* , *options*`)` macro automatically draws Boolean expressions in function notation. The function tree is drawn, then a row or column of inputs, then the connections. The default result is on the left, a custom element at the top, and a tree of gates only is shown on the right. `[Alogix.m4]`.

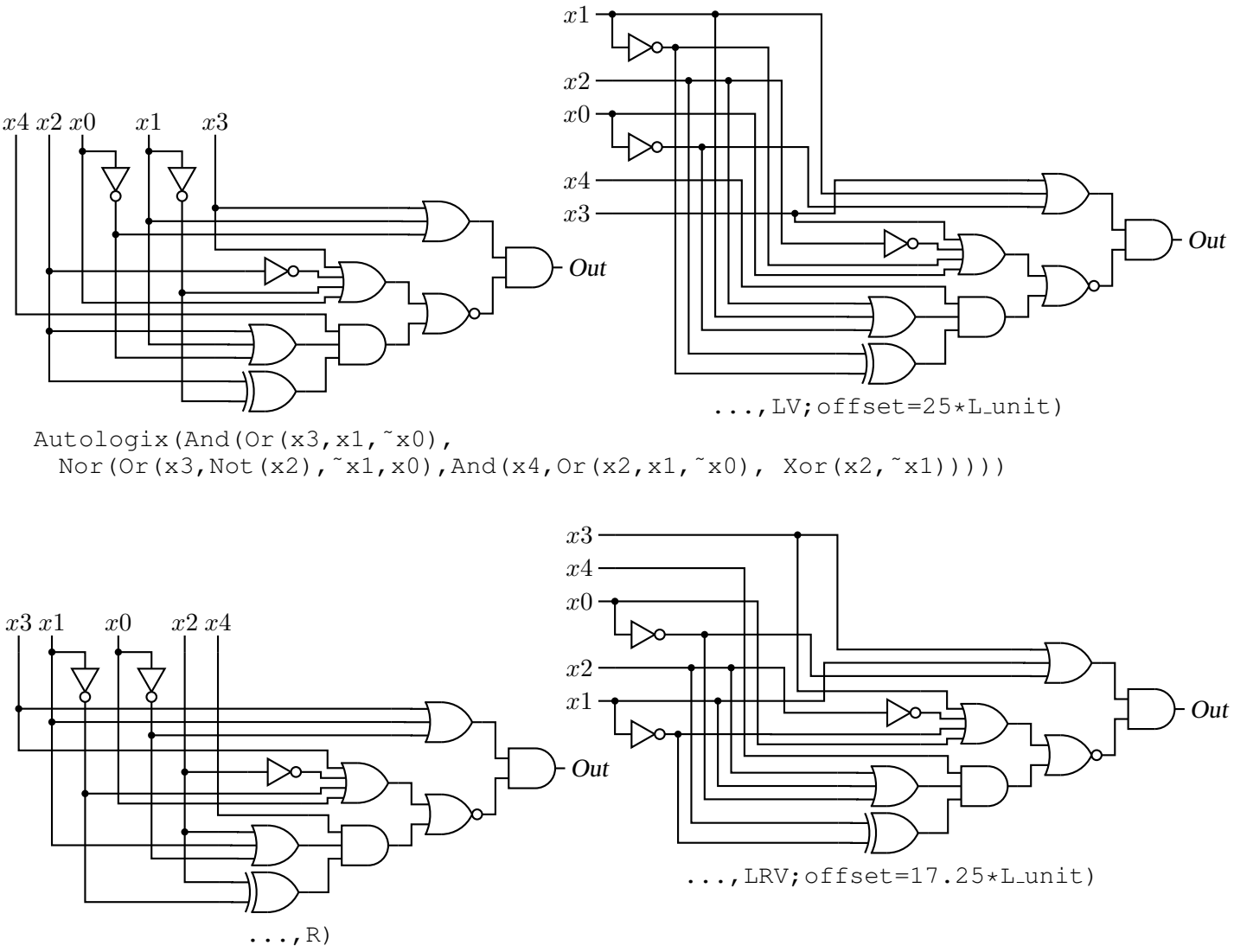


Figure 67: The `Autologix` macro can draw inputs on the left but the added drawing complexity may require hand tuning with second-argument options: `L` puts the inputs on the left, `R` reverses their order, `V` scans the input arguments in reverse order, and `offset=`*value* displaces the array of inputs `[ABlogix.m4]`.

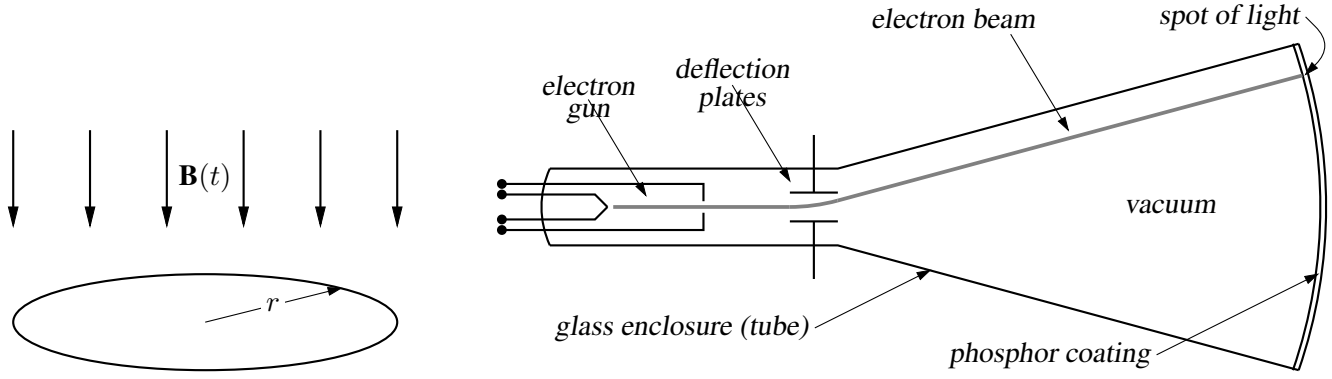


Figure 68: Line diagrams `[ex00.m4]`.

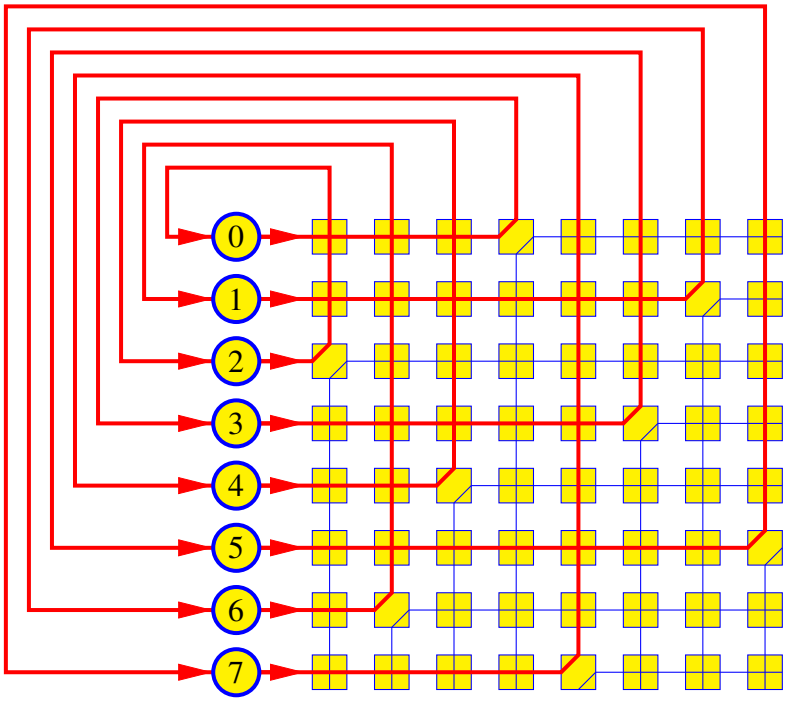


Figure 69: A crossbar switch `[Crossbar.m4]`.

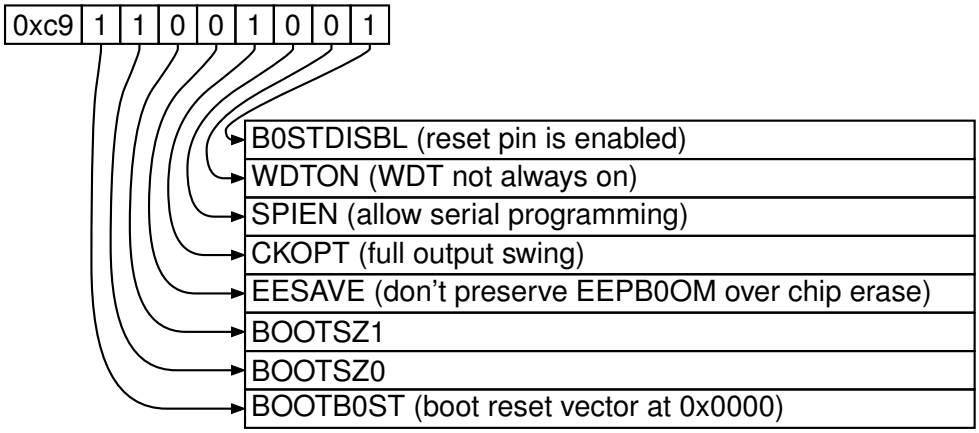


Figure 70: Elementary splines `[Byte.m4]`.

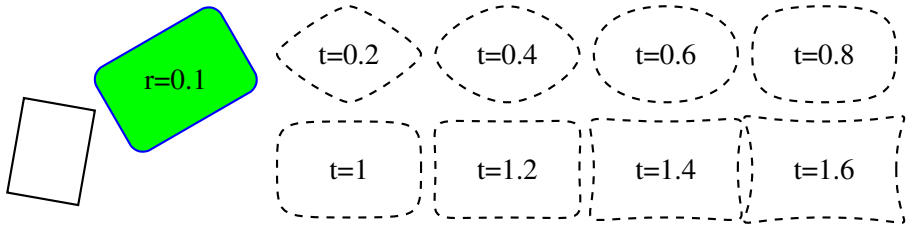


Figure 71: The macro `rotbox(`*wid,ht,type*`,[r|t=`*val*`])` draws a box in the current direction `[Rotbox.m4]`.

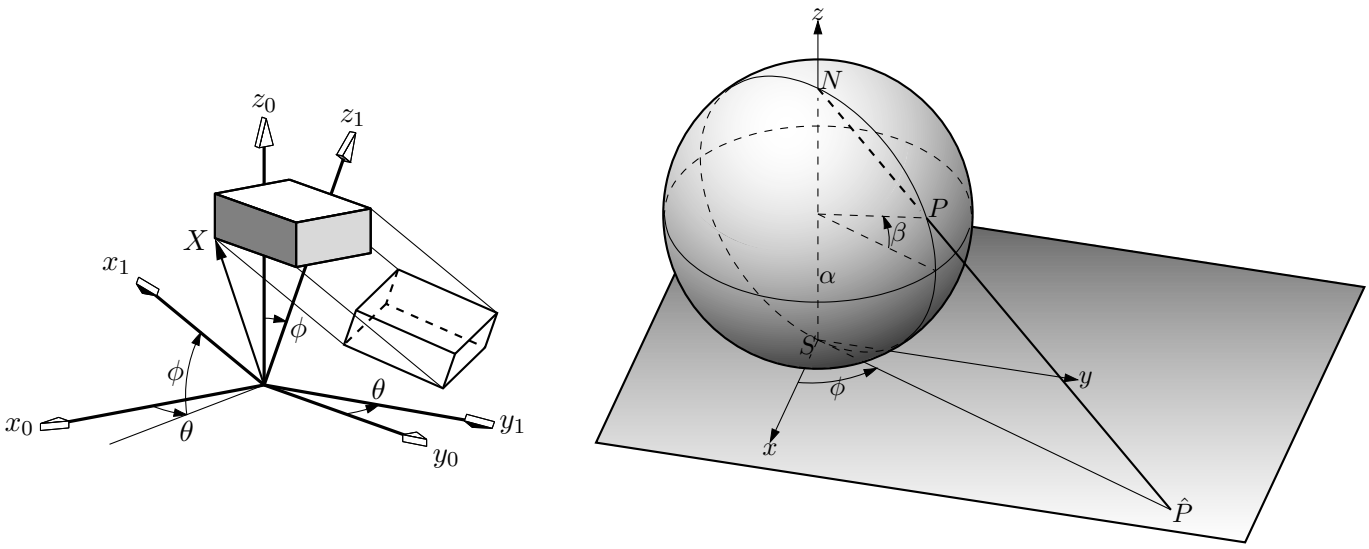


Figure 72: Test of `project` and other `lib3D` macros, showing the projection of a solid onto the y_1, z_1 plane by sighting along the x_1 axis. `[exp.m4]`.

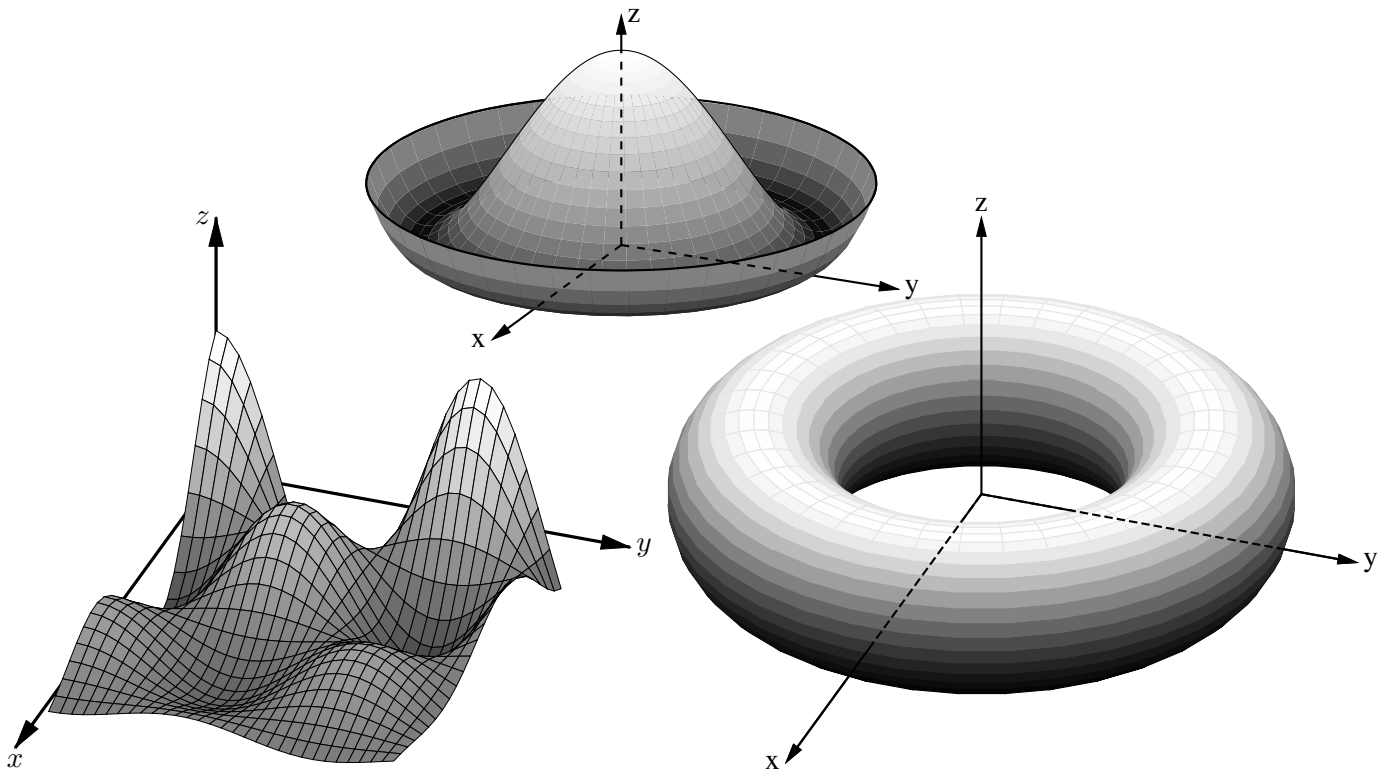


Figure 73: Plotting surfaces using gray scales. A sort algorithm determines plotting order `[graysurf.m4]`.

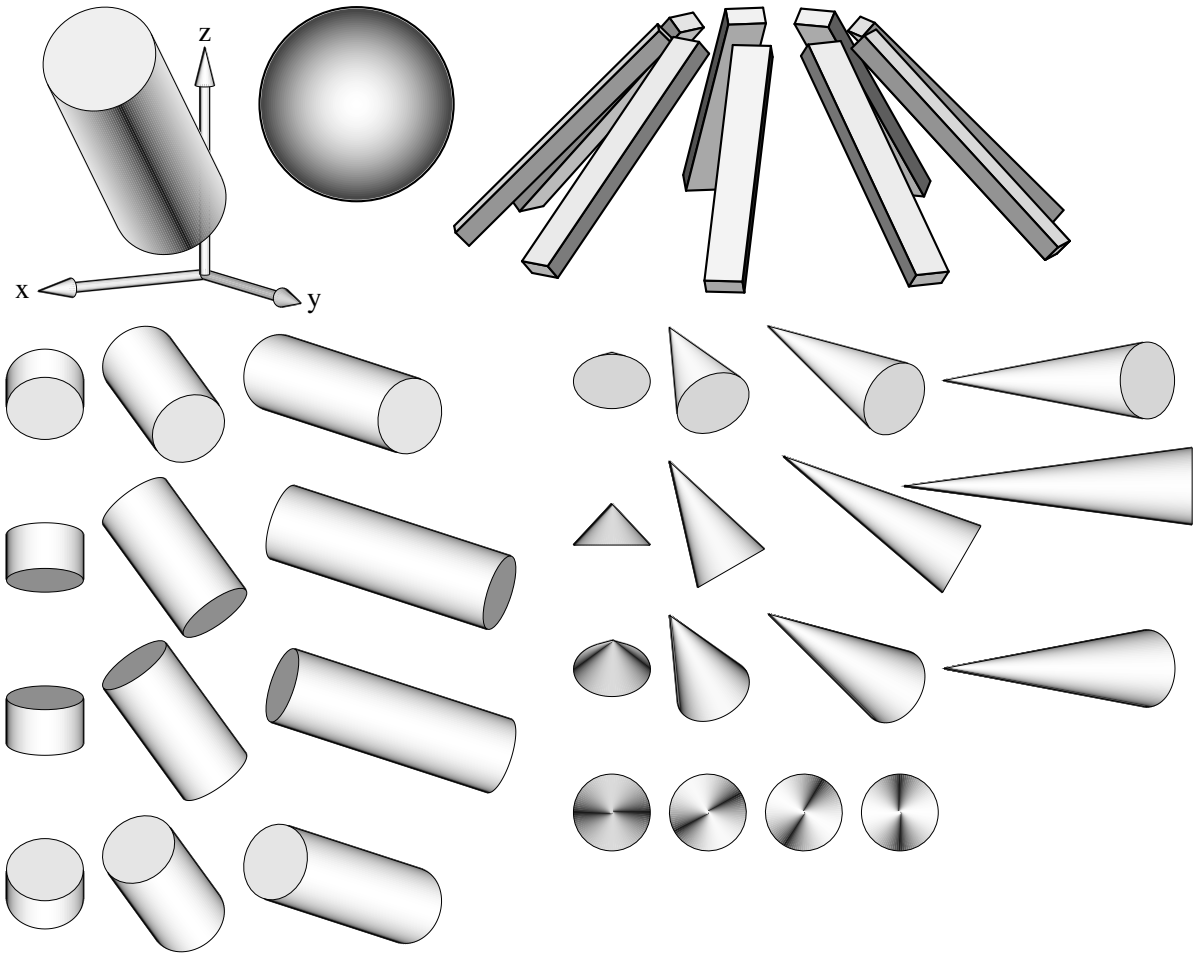
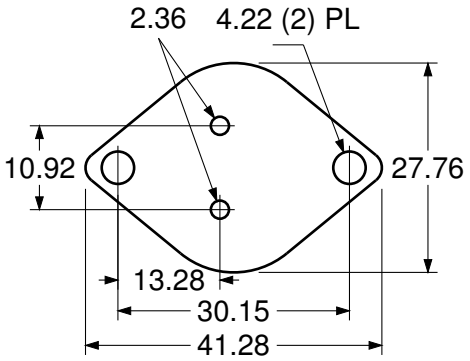
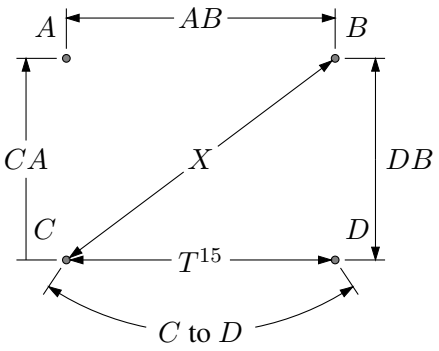
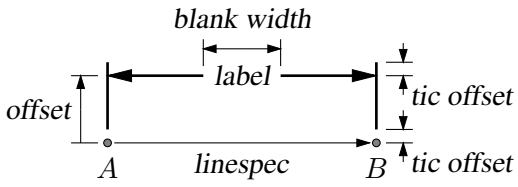


Figure 74: Basic shapes `[shapes.m4]`.








```
dimension_(from A to B,0.3,$AB$,20bp__)
dimension_(from C to B,,$X$,16bp__)
dimension_(from C to A,0.3,$CA$,14bp__,,->)
dimension_(from D to B,-0.3,"$DB$" ljust)
dimension_(from C to D,,s_box($T^{%g}$,15),W)
arcdimension_(from C to D with .c at \
  0.5 between A and B,12bp__,s_box($C$ to $D$),W)
dimension_(from A to B,0.5,\sl label,29bp__,0.1)
```

Figure 75: Illustrating the macro `dimension_(`*linespec*, *offset*, *label*, `D|H|W|`*blank width*, *tic offset*,`<-|->)`. A negative second argument implies an offset to the right of the *linespec* direction. A *label* starting with `"` or `sprintf` is copied literally. If *label* is an `s_box(...)` then setting argument 4 to `H`, `W`, or `D` tailors the blank width to the `s_box` height, width, or diagonal respectively; i.e., `W` is equivalent to `s_wd+textoffset*2`. The macro `arcdimension_` is similar but the first argument specifies the arc to be dimensioned and the second argument is the outward radial offset of the dimension arrow arc. `[ex09.m4]`.

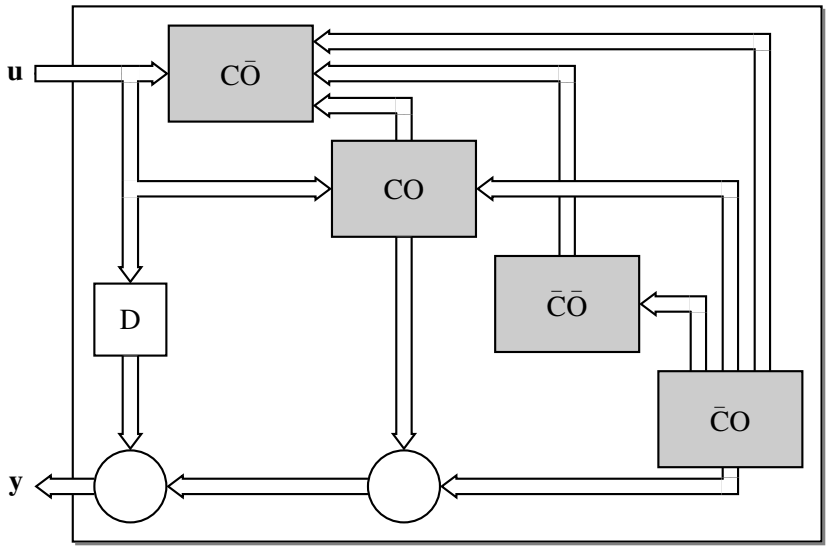
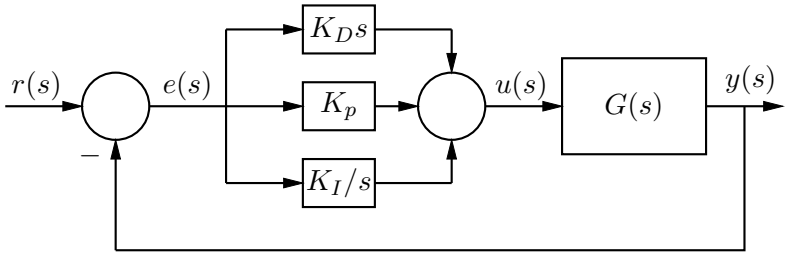
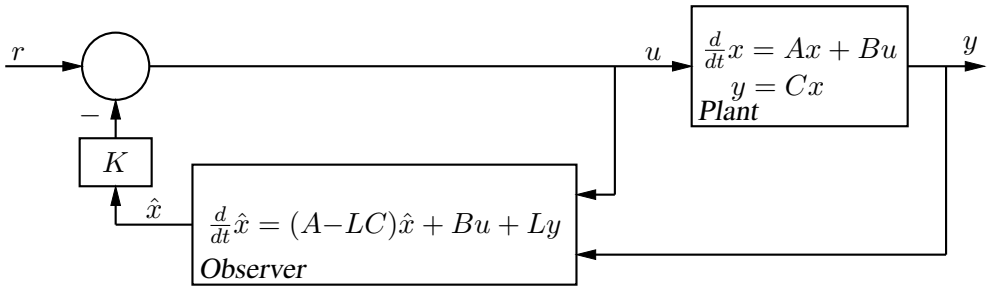


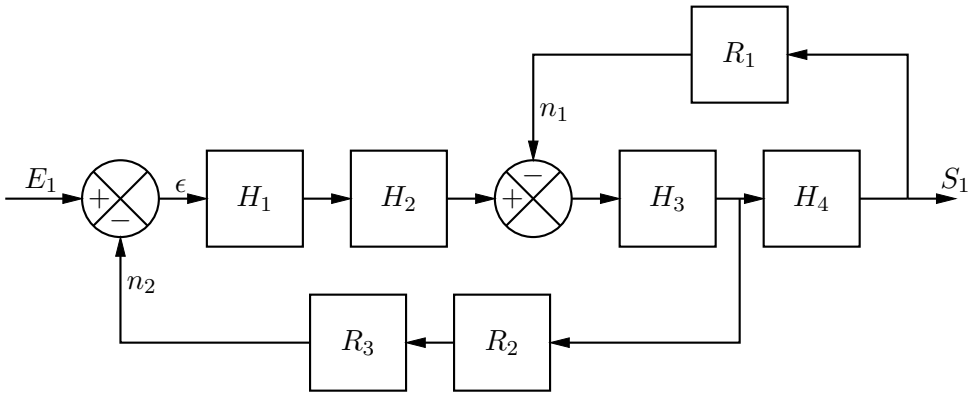
Figure 76: Use of `darrow` `[ex05.m4]`.



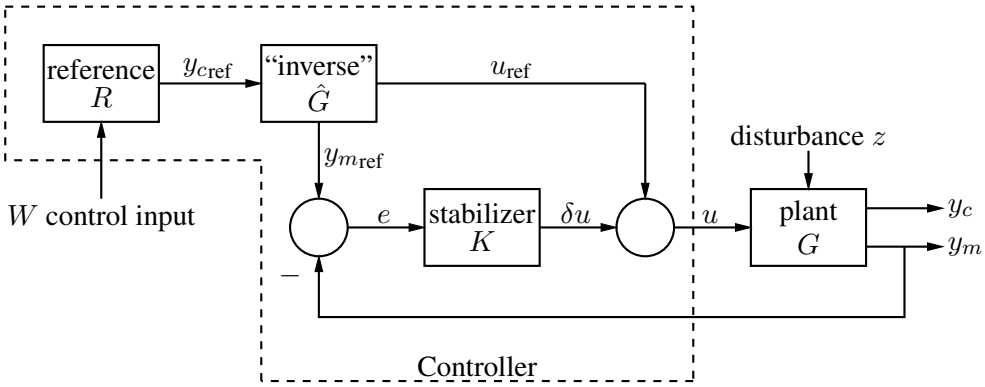
(a) PID control



(b) Output feedback with a full-order observer



(c) A multiblock example



(d) Nonlinear feedforward (for performance) and small-signal feedback (for stability)

Figure 77: Control-system block diagrams that do not require m4 `[control.m4]`.

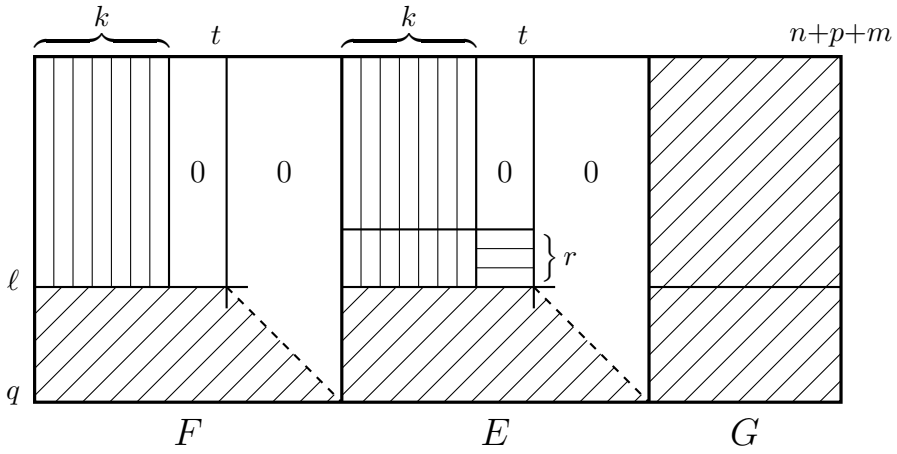


Figure 78: Crosshatching by `for` loops `[ex06.m4]`.

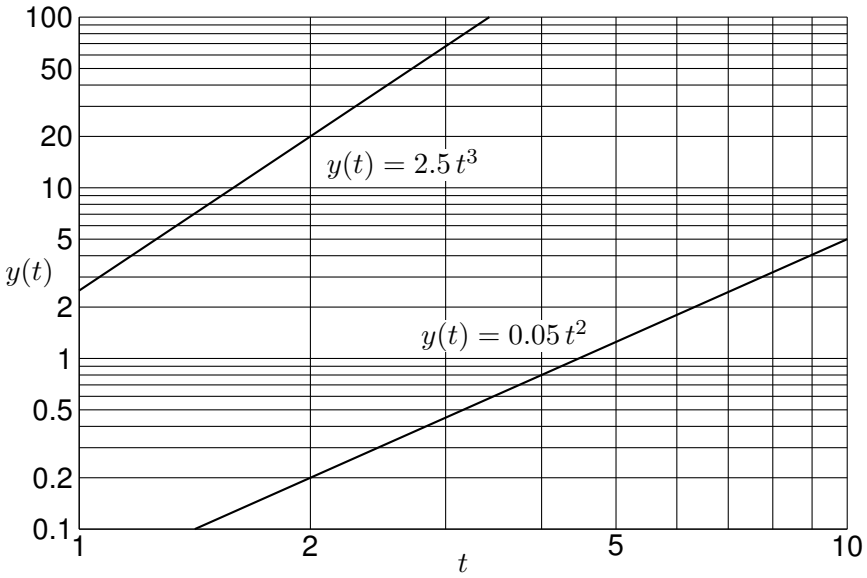


Figure 79: A graph drawn using the pic language `[Loglog.m4]`.

Figure 80: Conestoga Sailing Club (illustrating the filling of arbitrary shapes) `[csc.m4]`.

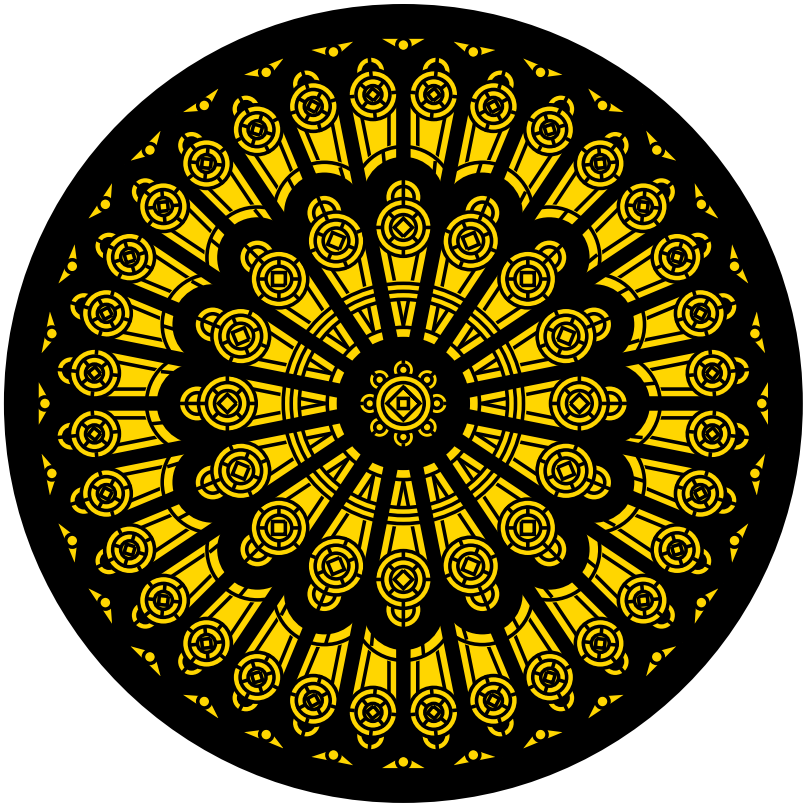


Figure 81: Redrawn from a detail of the set design for the musical *Dracula,* used for testing `dpic`. This diagram consumes much LaTeX main memory but can be produced directly as pdf, svg, or postscript by `dpic -d`, `dpic -g`, or `dpic -r` respectively since no text formatting is required `[rose.m4]`.

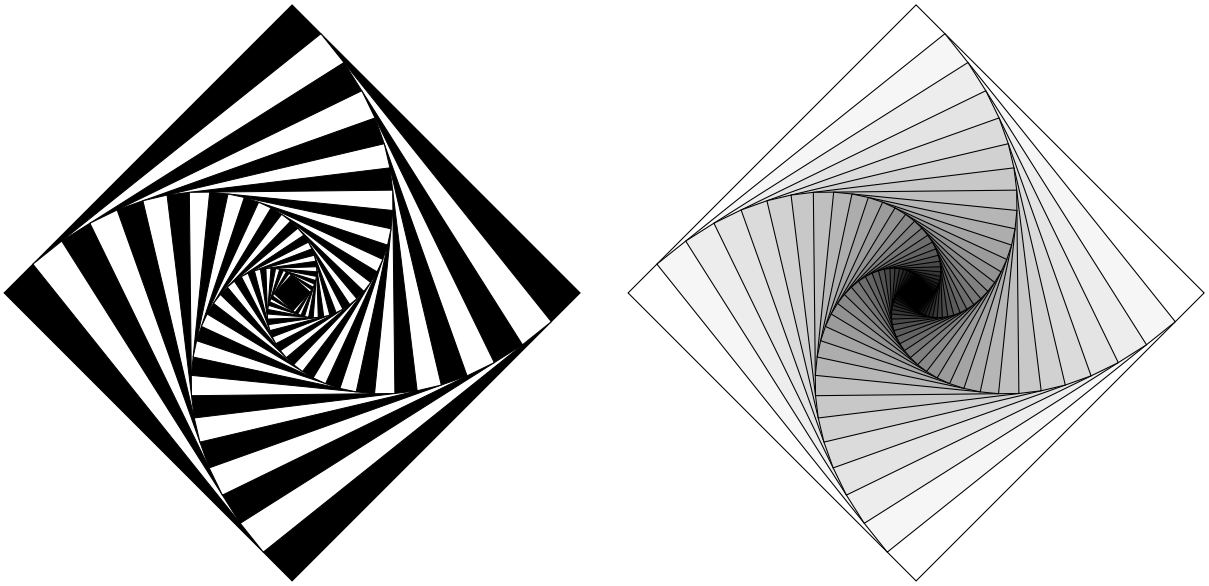


Figure 82: Variations on M. Goossens, S. Rahtz, and F. Mittelbach, *The LaTeX Graphics Companion,* Addison-Wesley 1997, pp. 57-58 `[diamond.m4]`.

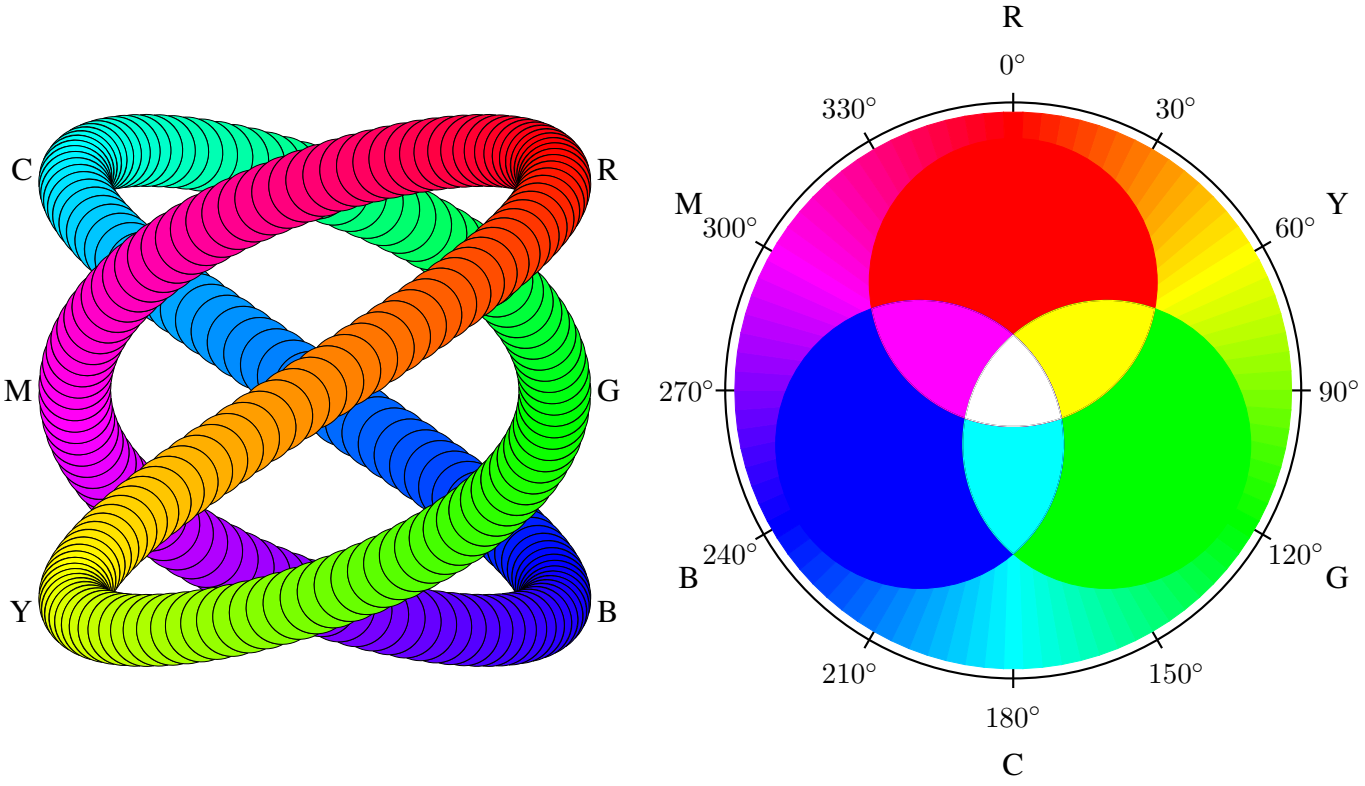


Figure 83: An exercise in calculating RGB colours `[worm.m4]`.

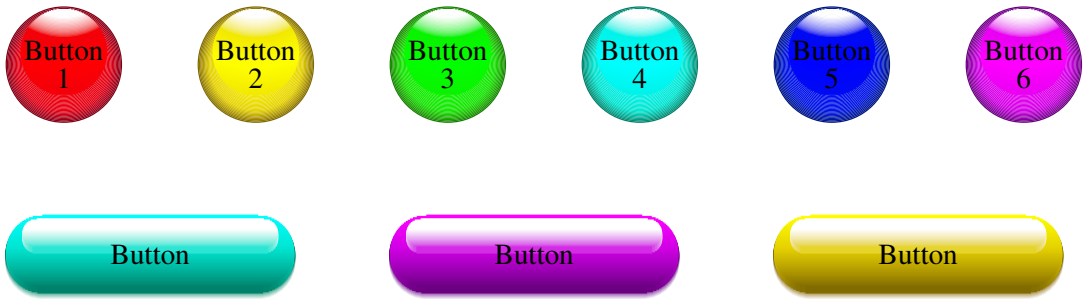


Figure 84: Shading in color `[Buttons.m4]`.

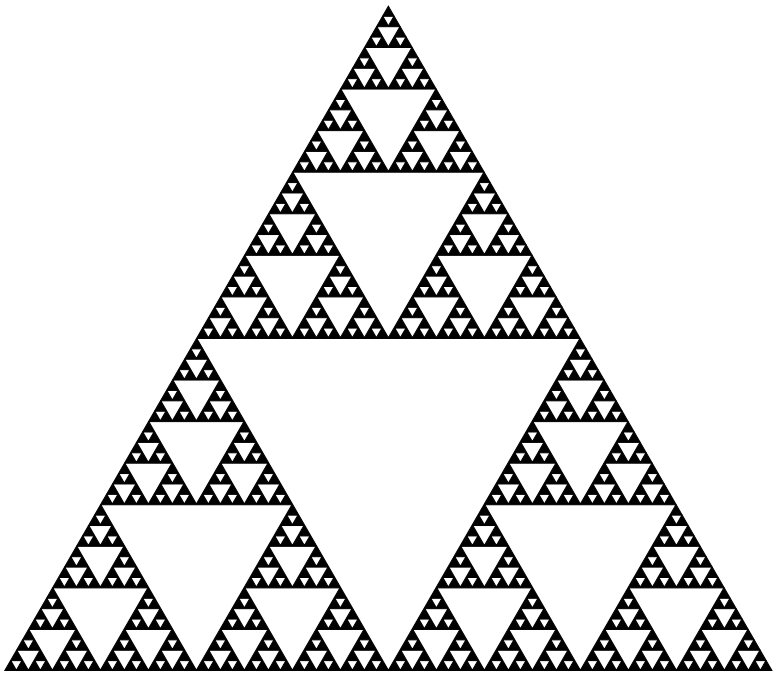


Figure 85: The Sierpinski triangle: a test of pic macro recursion `[Sierpinski.m4]`.

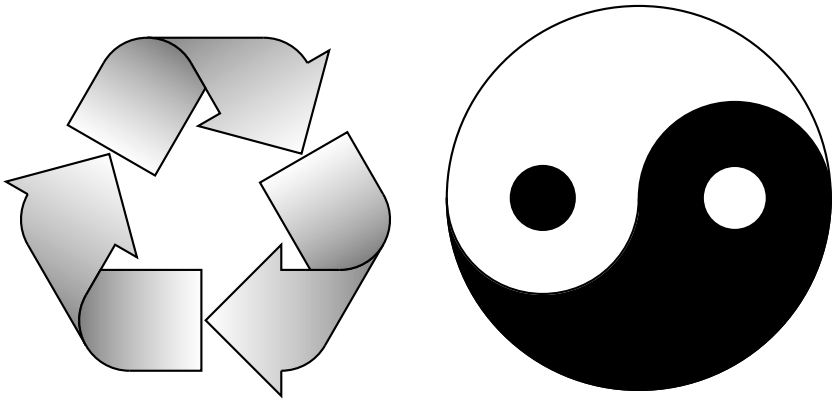


Figure 86: Modest repetition and partial fill `[recycle.m4]`.

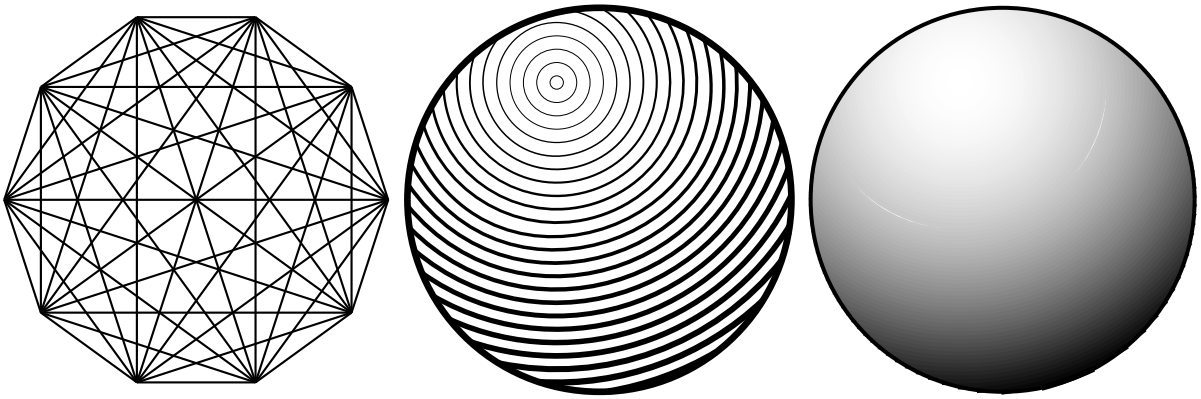


Figure 87: Simple diagrams that are easily drawn by looping `[ex15.m4]`.

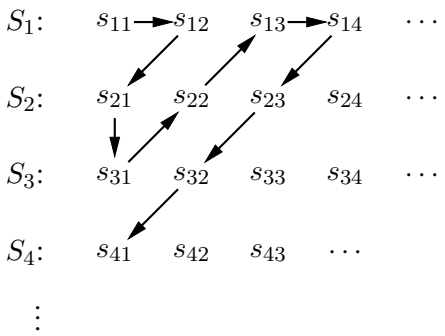


Figure 88: An example of enumeration `[Counting.m4]`.

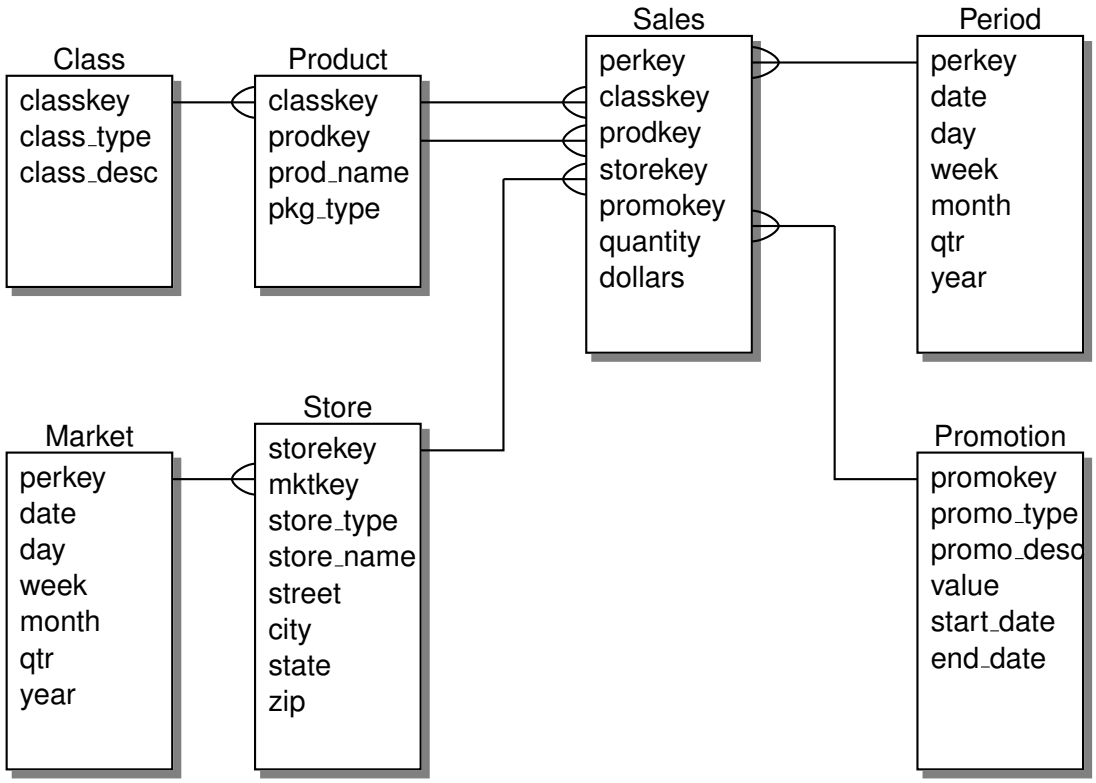


Figure 89: Illustrating `shadebox` and a custom crowfoot line termination `[Crow.m4]`.

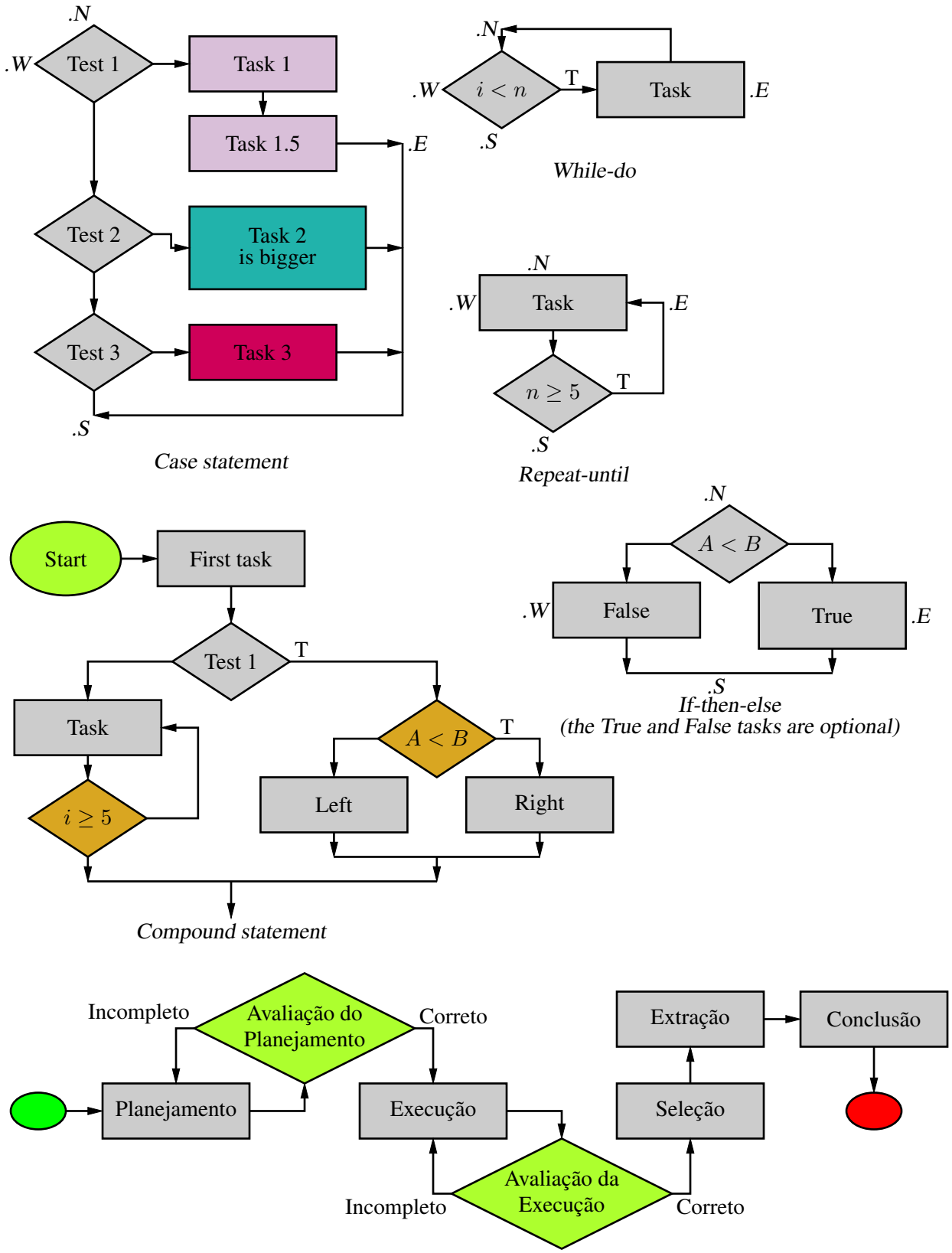


Figure 90: A flowchart sampler `[Flow.m4]`.

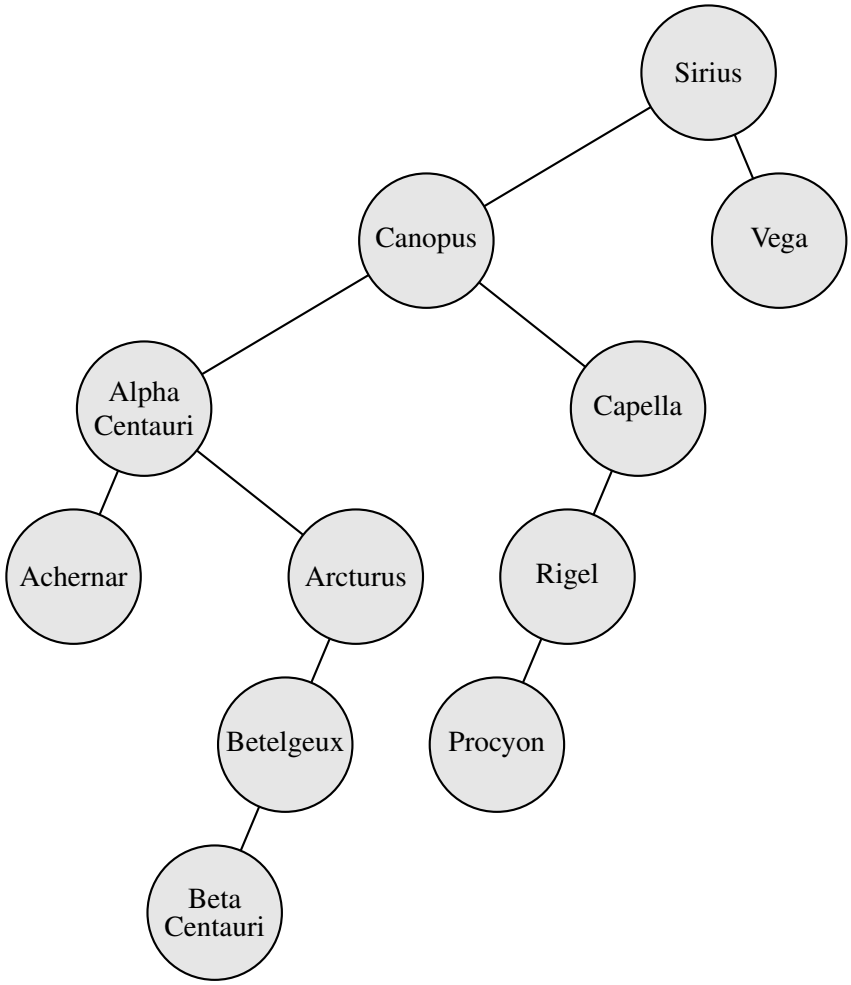


Figure 91: A binary tree `[Btree.m4]`.

Figure 92: Overlaying a figure with line graphics `[Incleps.m4]`.