Examples: Version 6.7

This is a collection of diagrams the author has had occasion to produce using m4 circuit macros and others, and gpic or dpic. In some cases, there are other or better m4 or pic constructs for producing the same drawings, but the source-file names are shown for reference. Most of the examples can be processed using either dpic -p or gpic -t, which has meant that in a few cases the source is slightly more complicated than it would be if only one processor had been assumed.

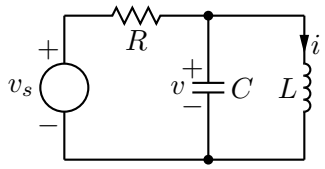


Figure 1: The quick-start example from the manual [quick.m4].

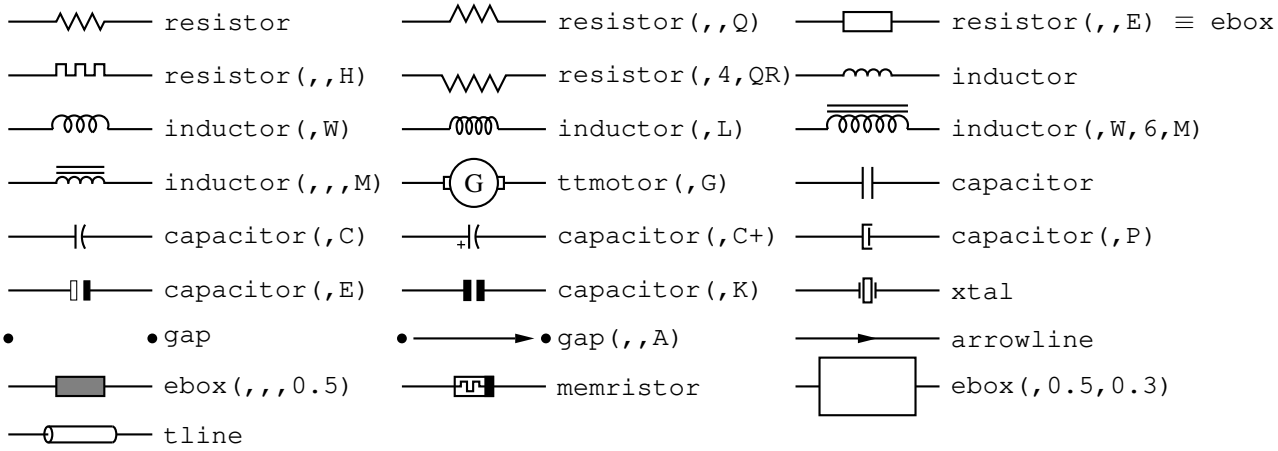


Figure 2: Two-terminal elements, showing some variations [CctTable.m4].

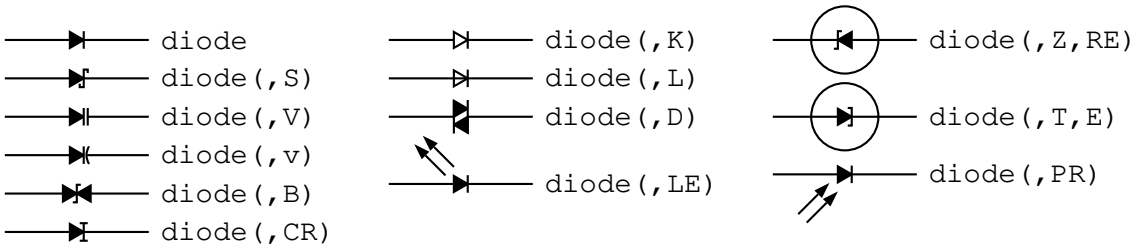


Figure 3: Diodes [Diodes.m4].

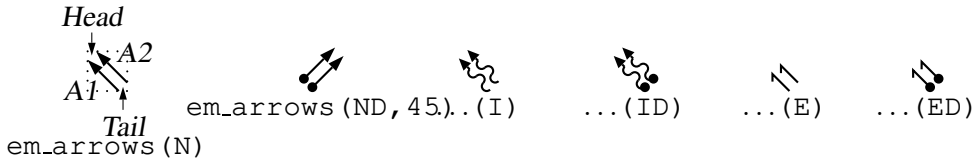


Figure 4: Radiation arrows [Emarrows.m4].

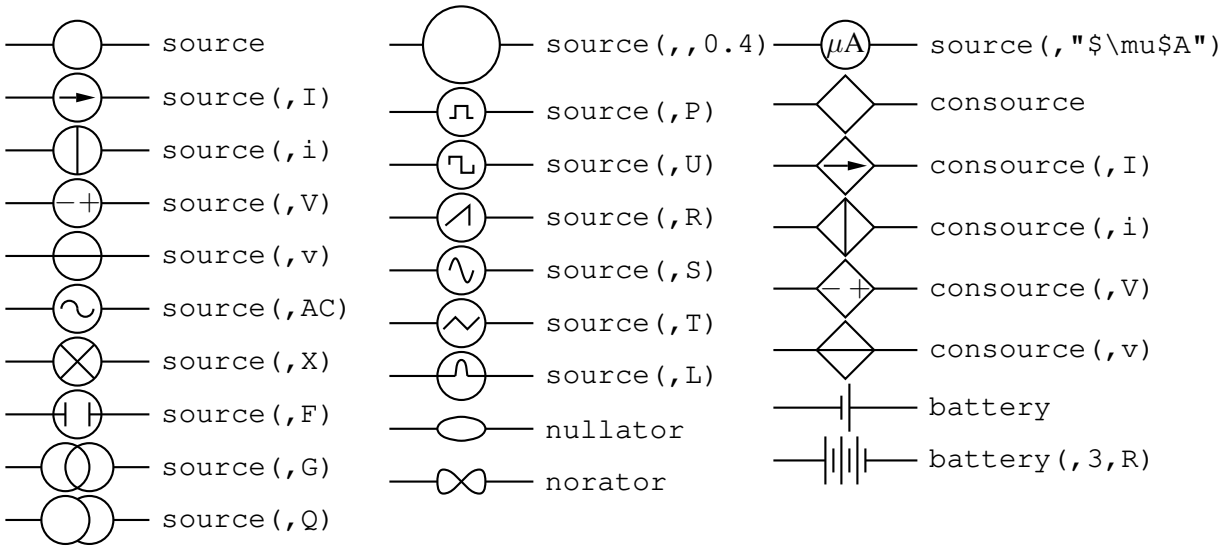


Figure 5: Sources and source-like elements `[Sources.m4]`.

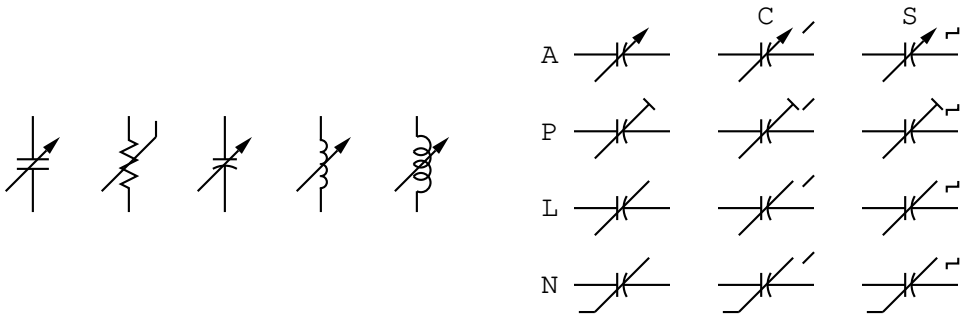


Figure 6: Arrows and marks indicating variability `[Variable.m4]`.

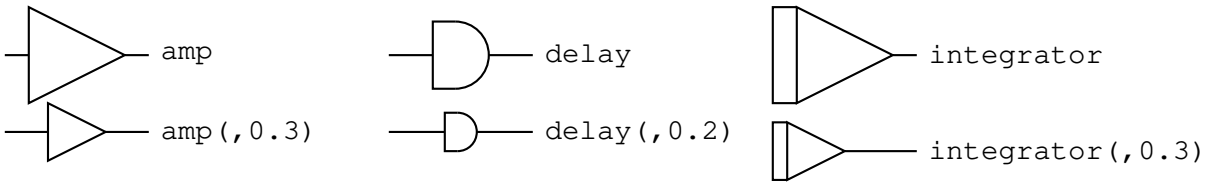


Figure 7: Macros `amp`, `delay`, and `integrator` `[AmpTable.m4]`.

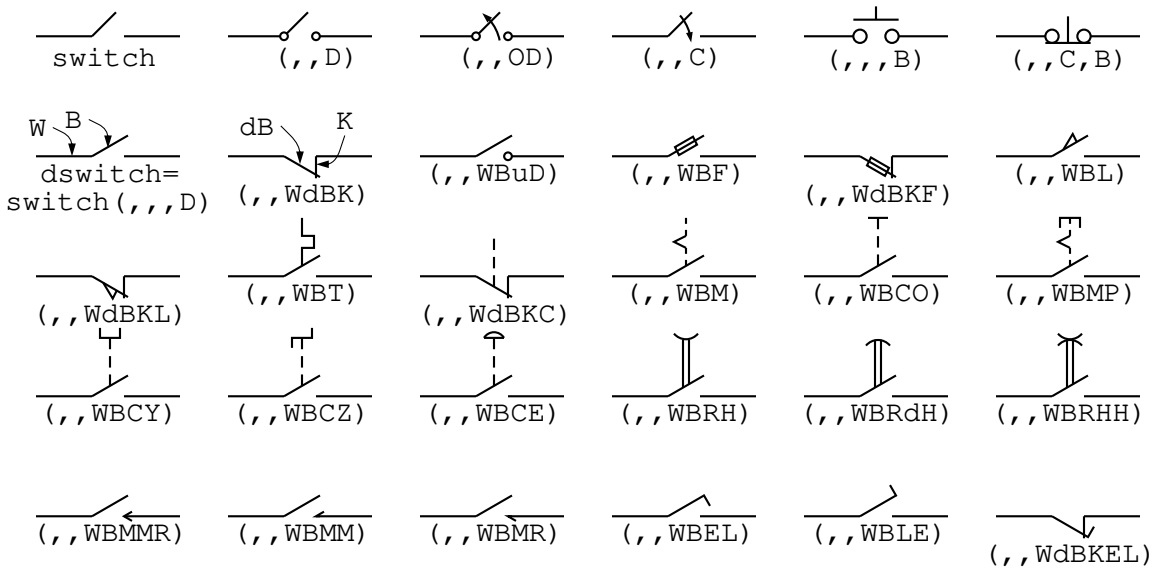


Figure 8: The `switch` and `dswitch` macros `[Switches.m4]`.

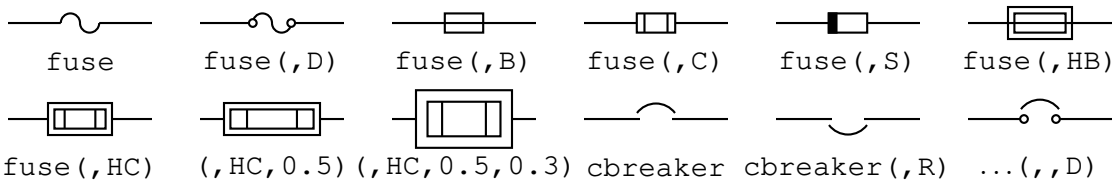


Figure 9: Macros `fuse` and `cbreaker` `[Fuses.m4]`.

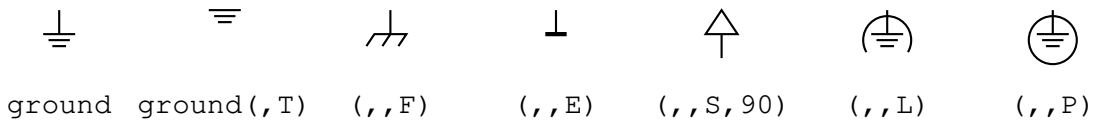


Figure 10: Ground symbols `[Grounds.m4]`.

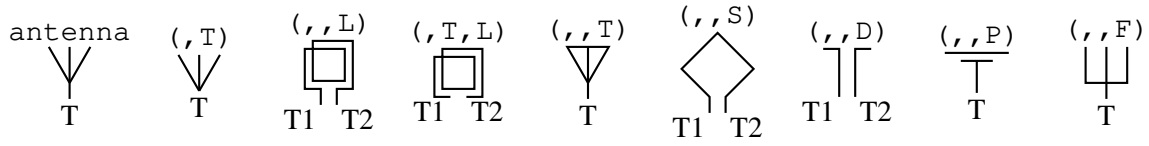


Figure 11: Antenna symbols `[Antennas.m4]`.

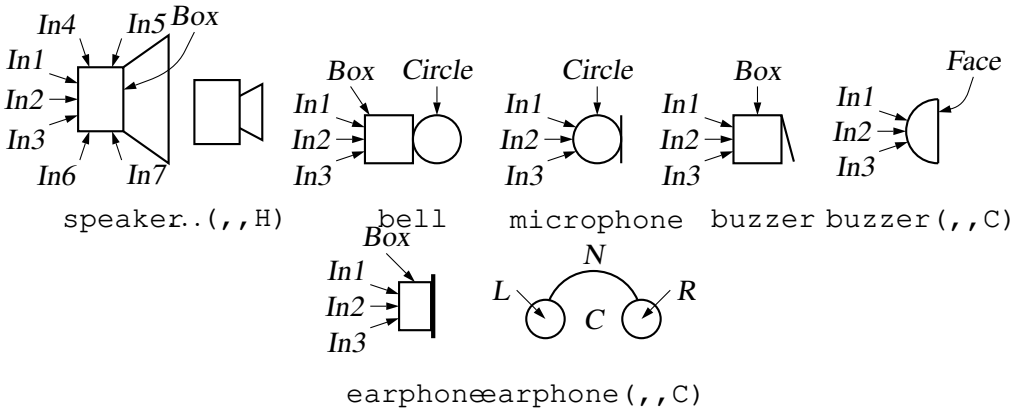


Figure 12: Audio elements [Audio.m4].

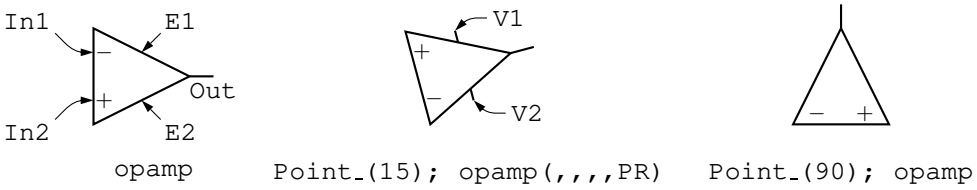


Figure 13: The opamp [Opamp.m4].

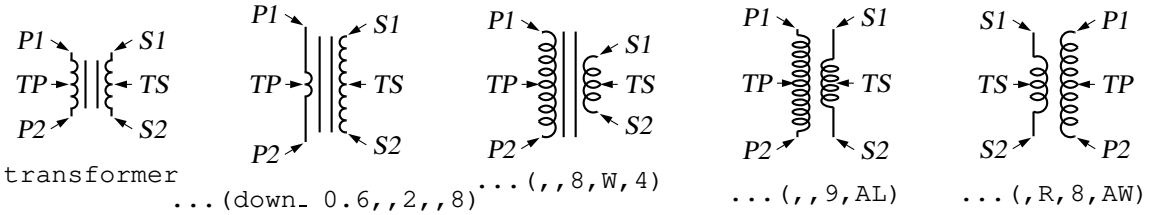


Figure 14: The transformer element, drawing direction down [Xform.m4].

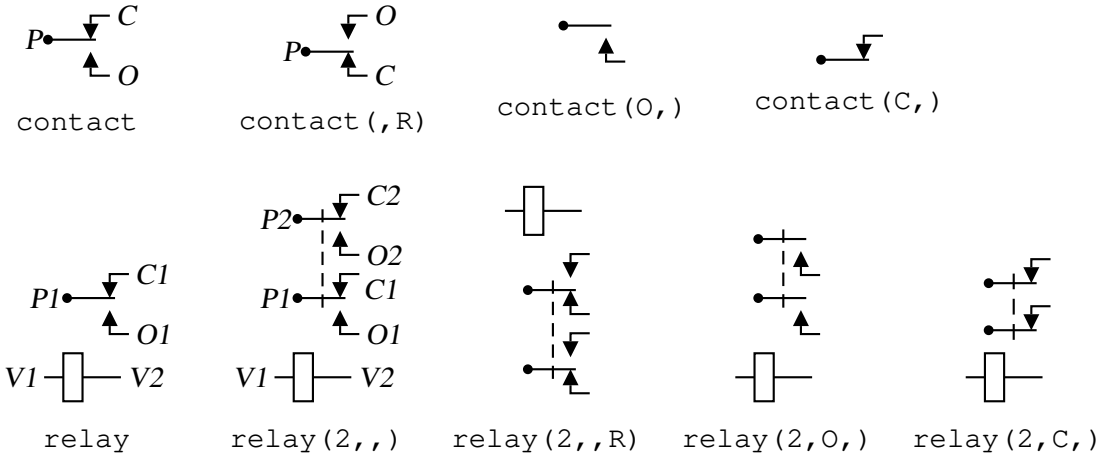


Figure 15: The contact and relay macros [Relay.m4].

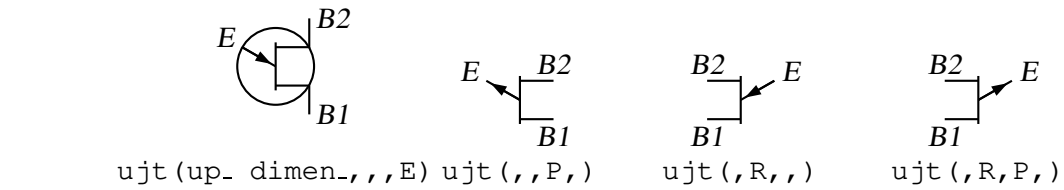


Figure 16: UJT examples `[ujt.m4]`.

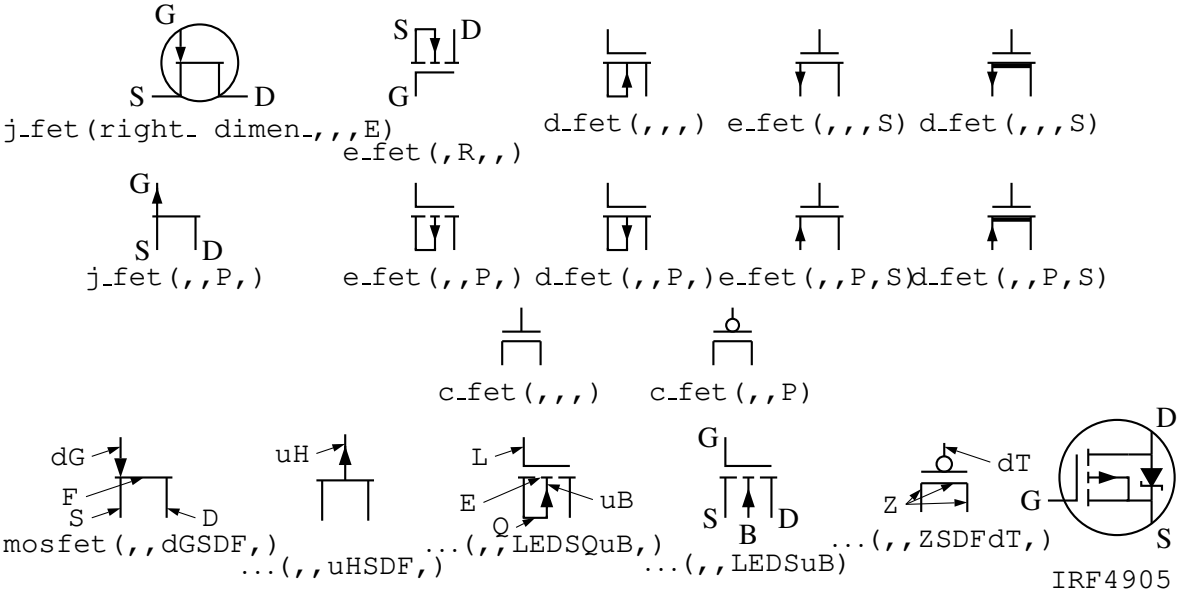


Figure 17: FETs, showing programmable components and example customizations `[fet.m4]`.

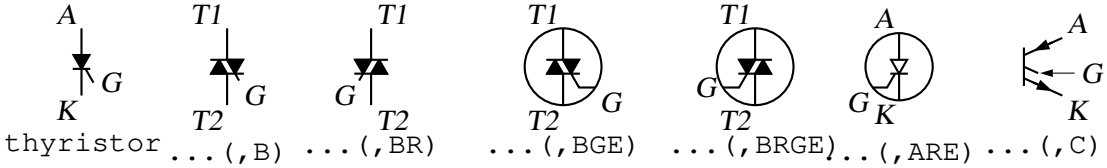


Figure 18: Thyristor examples `[thyristor.m4]`.

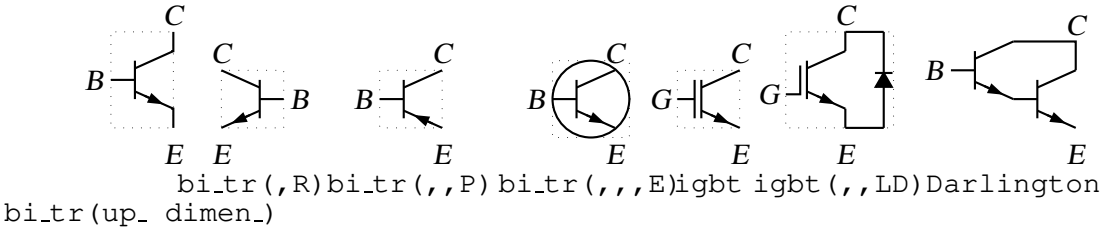


Figure 19: Bipolar transistors (drawing direction: up) `[Bip.m4]`.

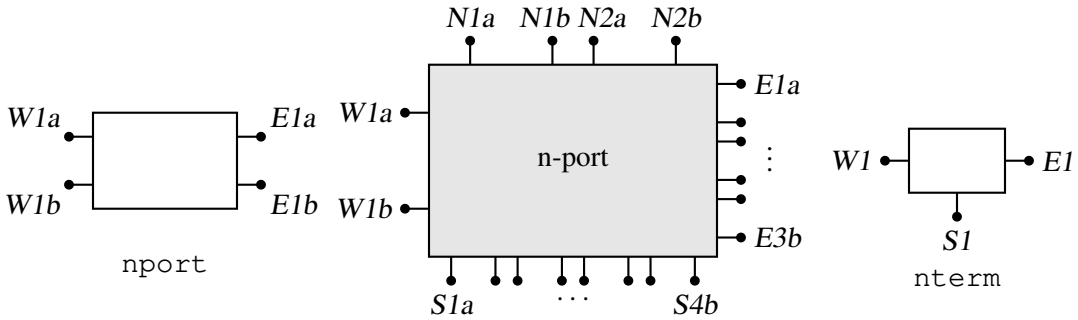


Figure 20: The `nport` and `nterm` macros `[Nport.m4]`.

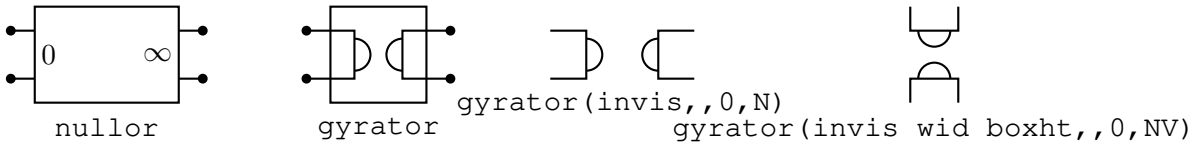


Figure 21: Some customizations of `nport` `[NLG.m4]`.

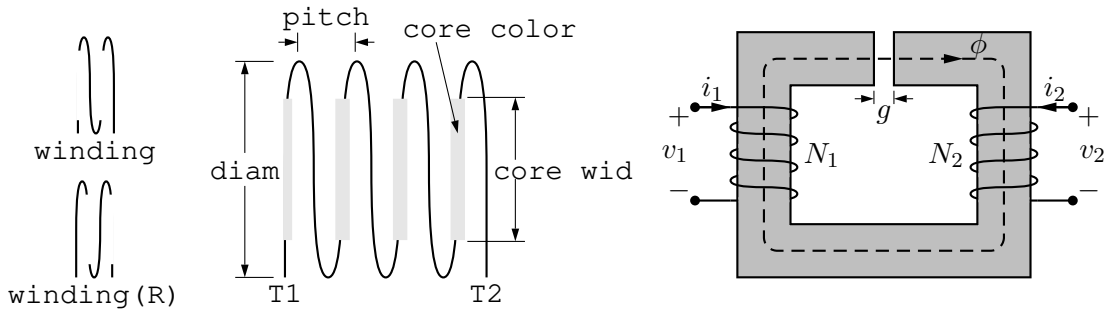


Figure 22: The macro `winding(L|R,diam,pitch,turns,core wid,core color)` `[Windings.m4]`.

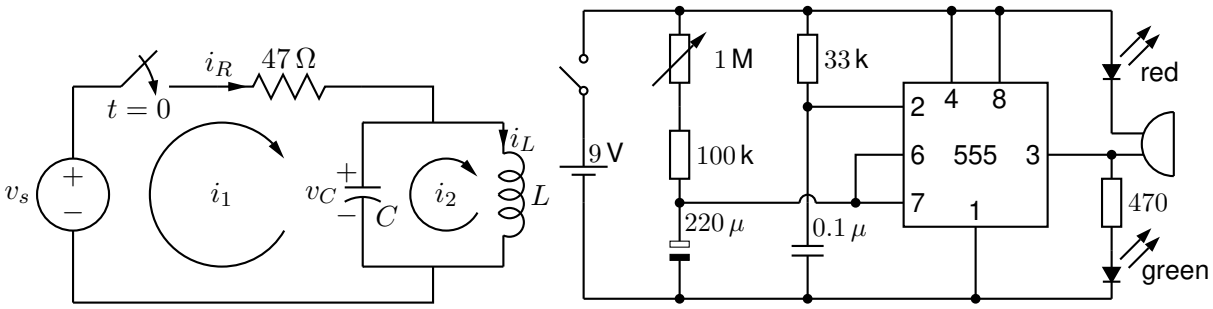


Figure 23: Two simple labeled circuits `[ex01.m4]` `[Timer.m4]`.

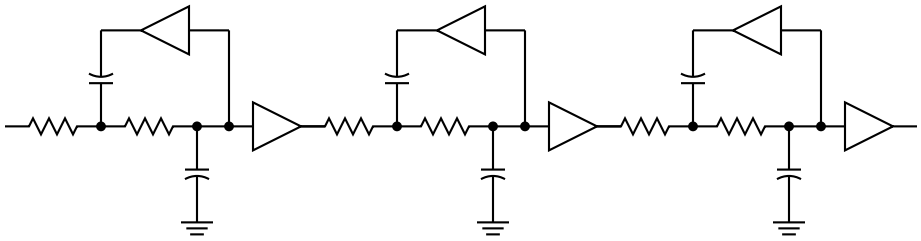


Figure 24: A six-pole filter `[Sixpole.m4]`.

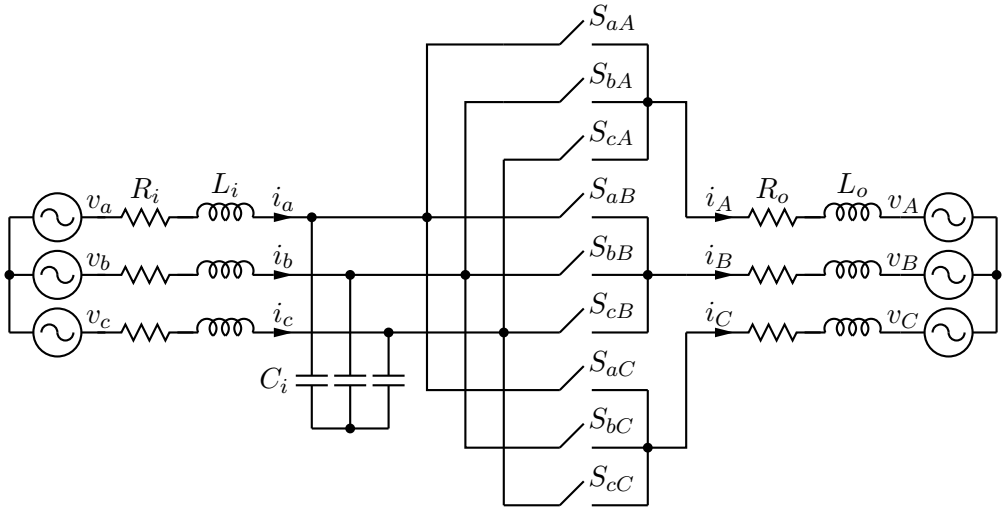


Figure 25: A three-phase switched AC-AC converter `[MC.m4]`.

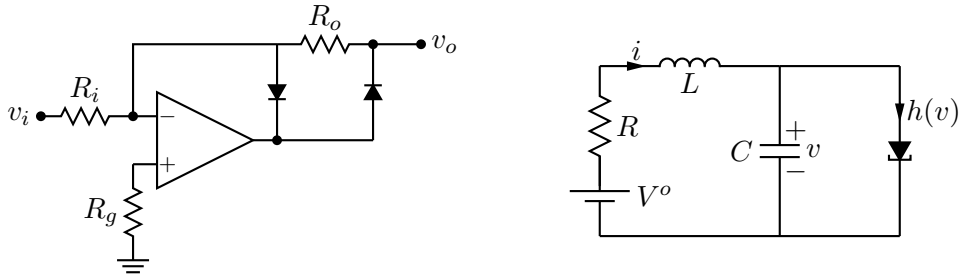


Figure 26: Precision half-wave rectifier and a tunnel diode circuit (illustrating `opamp`, `diode`, `resistor`, `ground`, and labels) `[ex18.m4][ex19.m4]`.

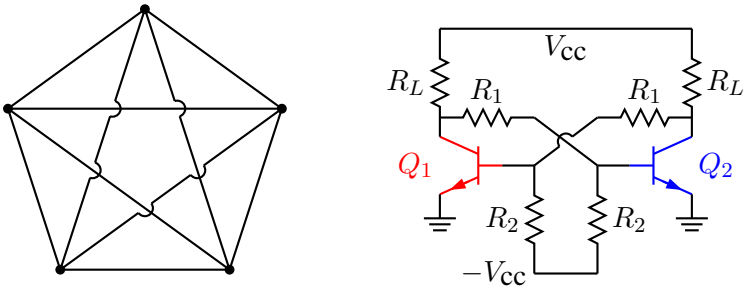


Figure 27: Non-planar graph and bistable circuit (illustrating the `crossover` macro and colored elements) `[ex10.m4][bistable.m4]`.

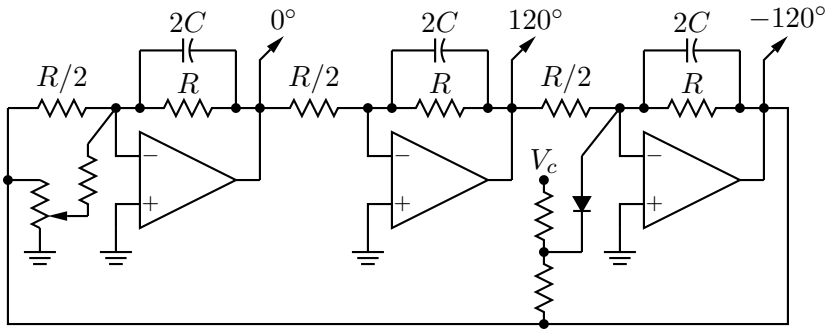


Figure 28: Three-phase oscillator `[Three.m4]`.

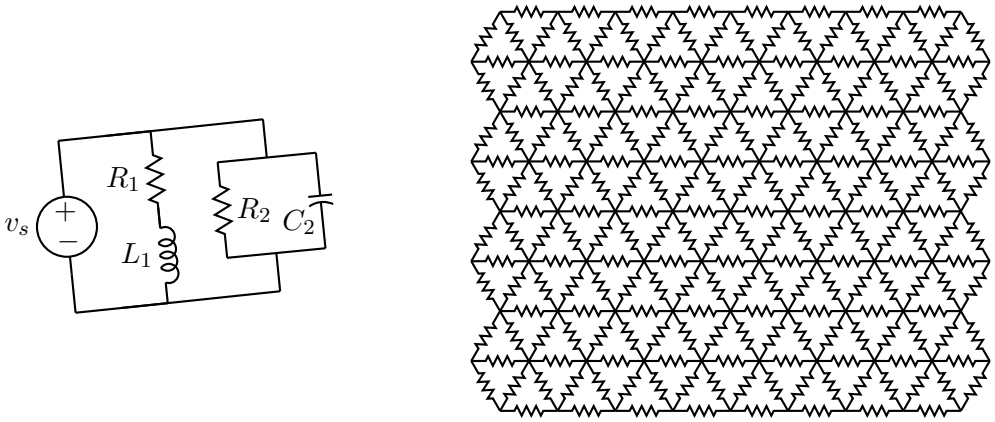


Figure 29: A skewed circuit used to test the macro `gpar_`(*element*, *element*, *separation*) , and a repetitive network created by Pic looping `[gpar.m4][ex17.m4]`.

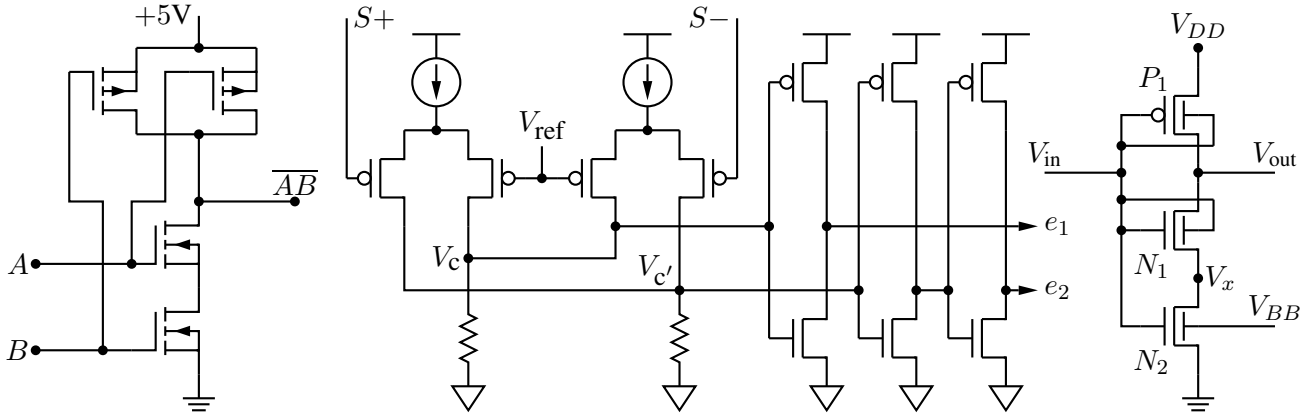


Figure 30: A CMOS NAND gate, a test circuit, and an XMOSFET example `[ex12.m4]`.

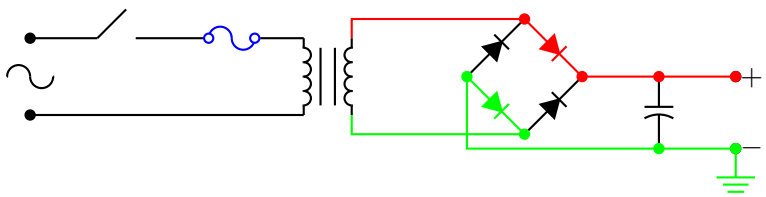


Figure 31: An elementary power supply circuit with colored elements `[pwrsupply.m4]`.

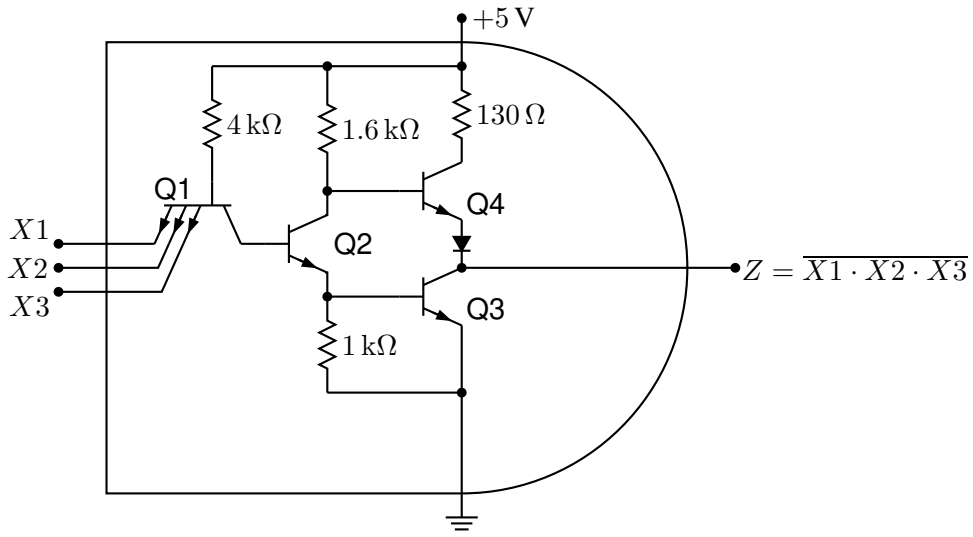


Figure 32: TTL NAND gate illustrating a transistor with multiple emitters `[TTLnand.m4]`.

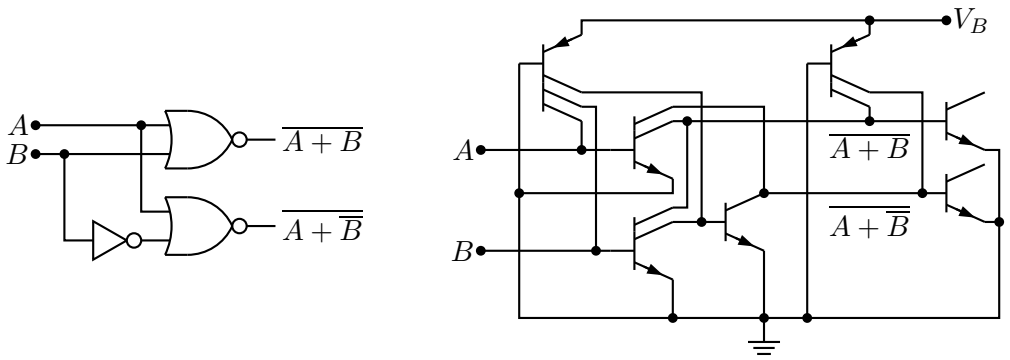


Figure 33: Gate circuit and equivalent embedded I^2L components illustrating multiple collectors `[I2L.m4]`.

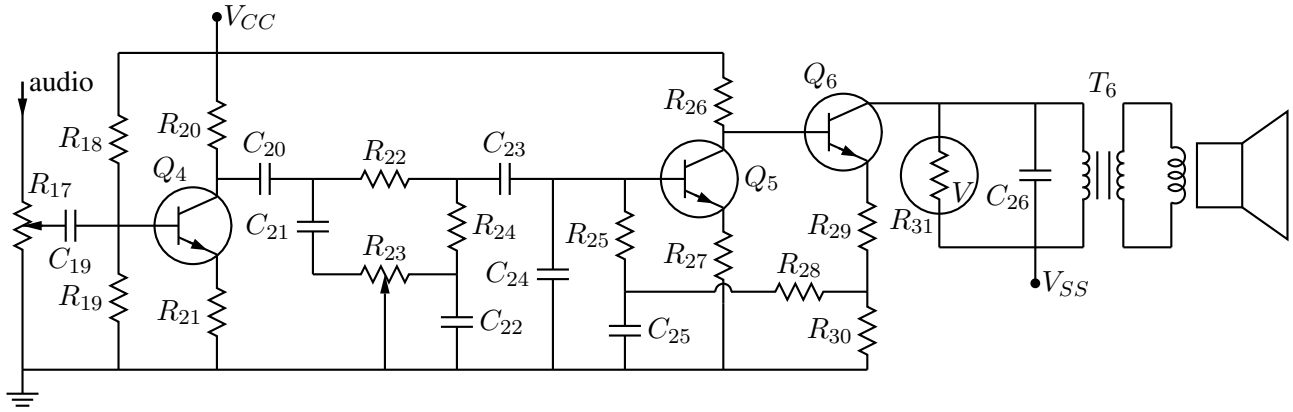


Figure 34: Transistor radio audio chain (a modest circuit with a few custom elements) `[ex11.m4]`.

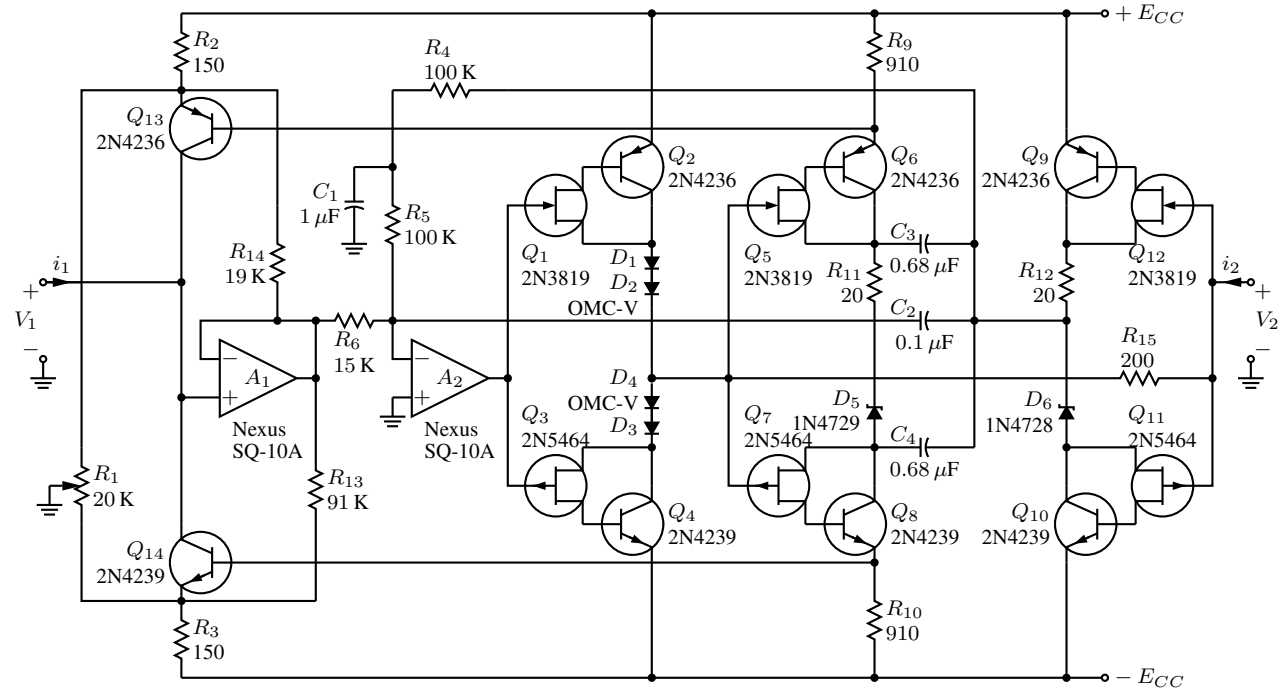


Figure 35: Realization of a controlled source (illustrating stacked element labels) `[Csource.m4]`.

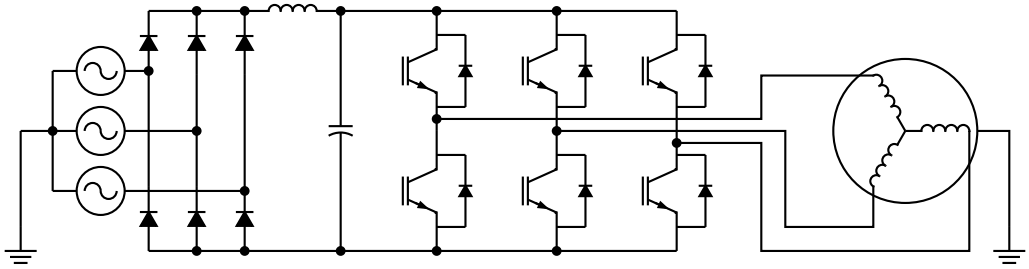


Figure 36: Synchronous machine driven by variable-speed drive and rectifier `[Drive.m4]`.

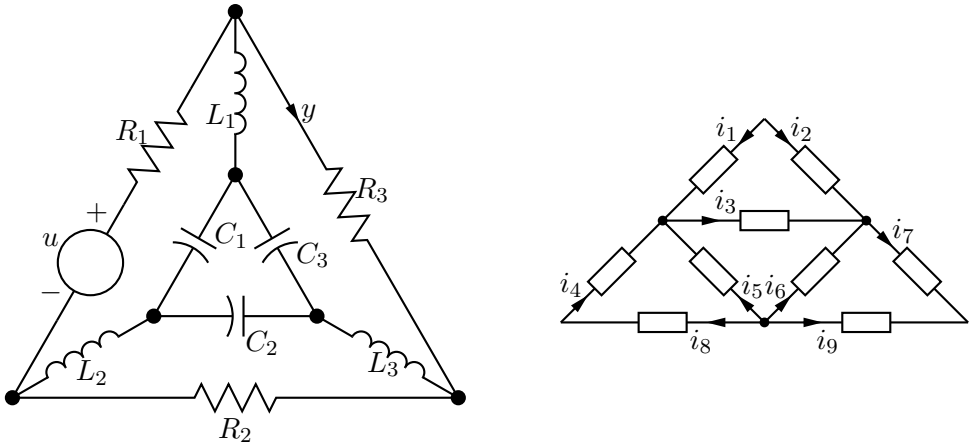


Figure 37: Labels on non-manhattan elements `[ex04.m4]`.

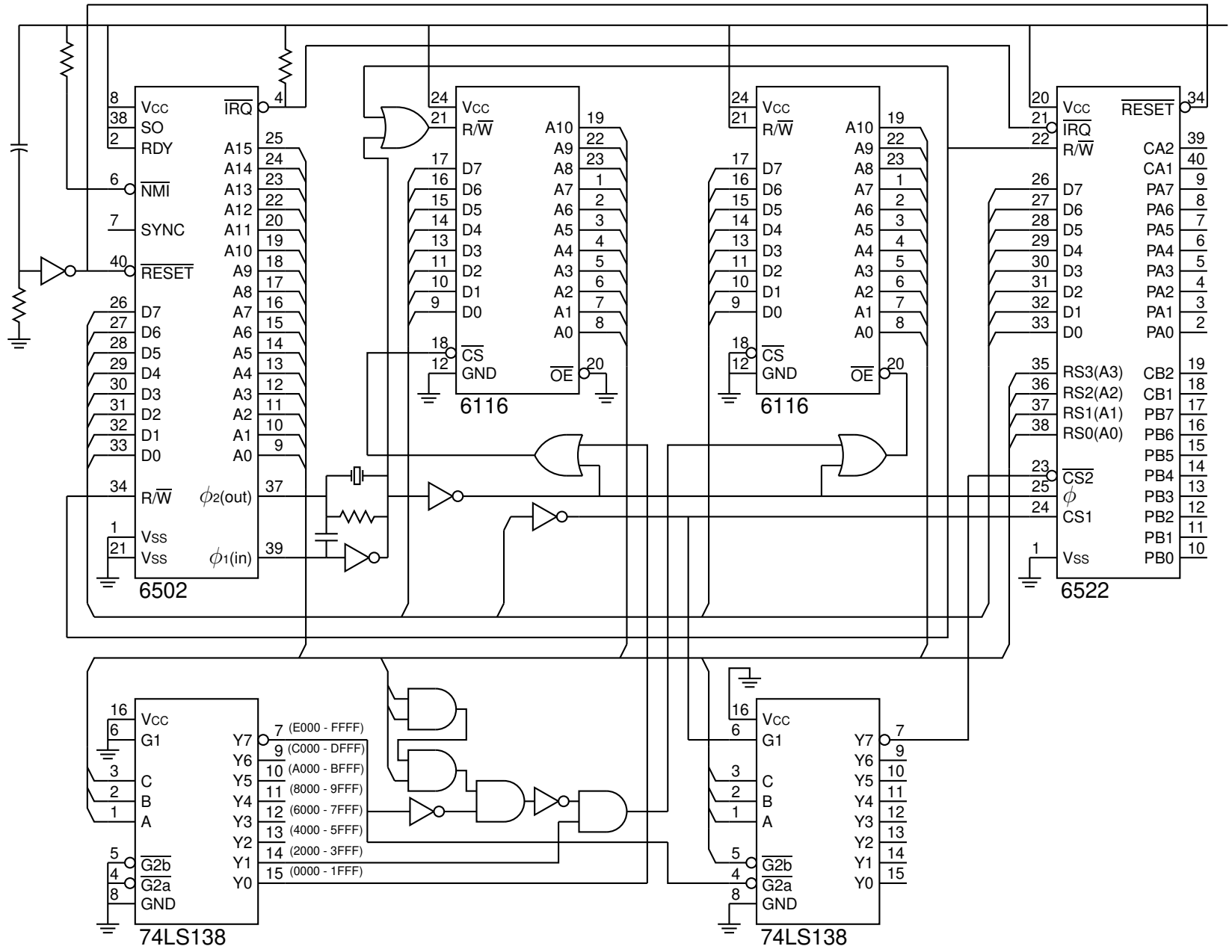


Figure 38: A digital circuit of moderate size, redrawn from M. P. Maclenan and G. M. Burns, “An Approach to Drawing Circuit Diagrams for Text Books,” Tugboat (12)1, March 1991, pp. 66-69 `[lcct.m4]`.

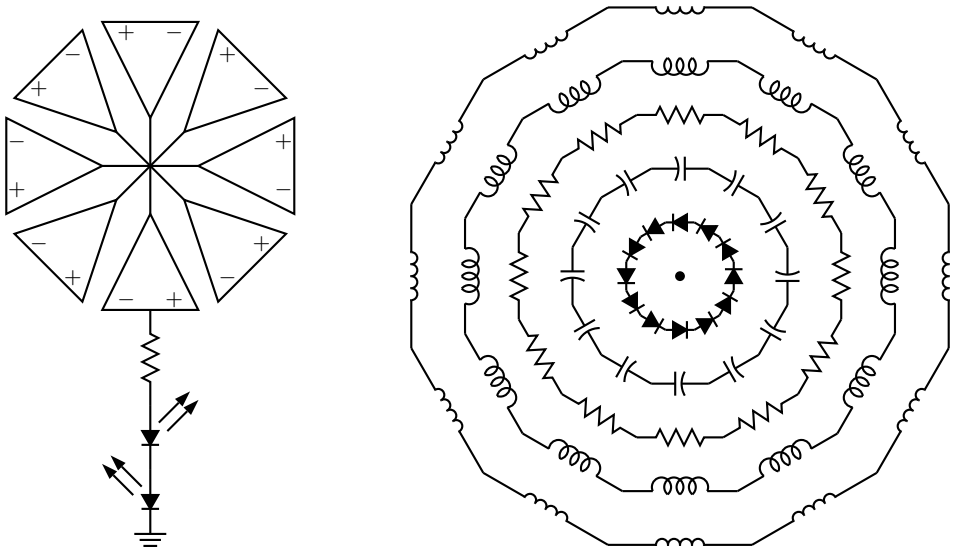


Figure 39: Elements at obtuse angles `[ex02.m4]`.

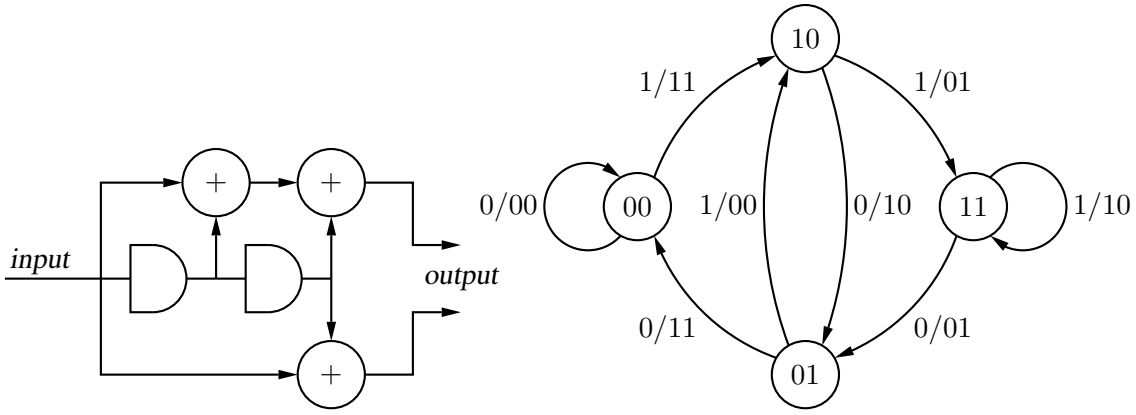


Figure 40: A rate $1/2$ binary convolutional coder and its state diagram `[ex16.m4]`.

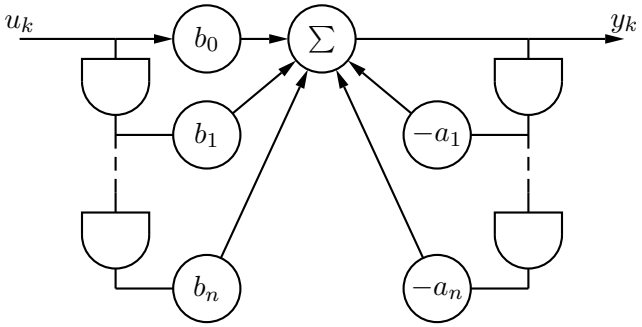


Figure 41: Digital filter `[ex03.m4]`.

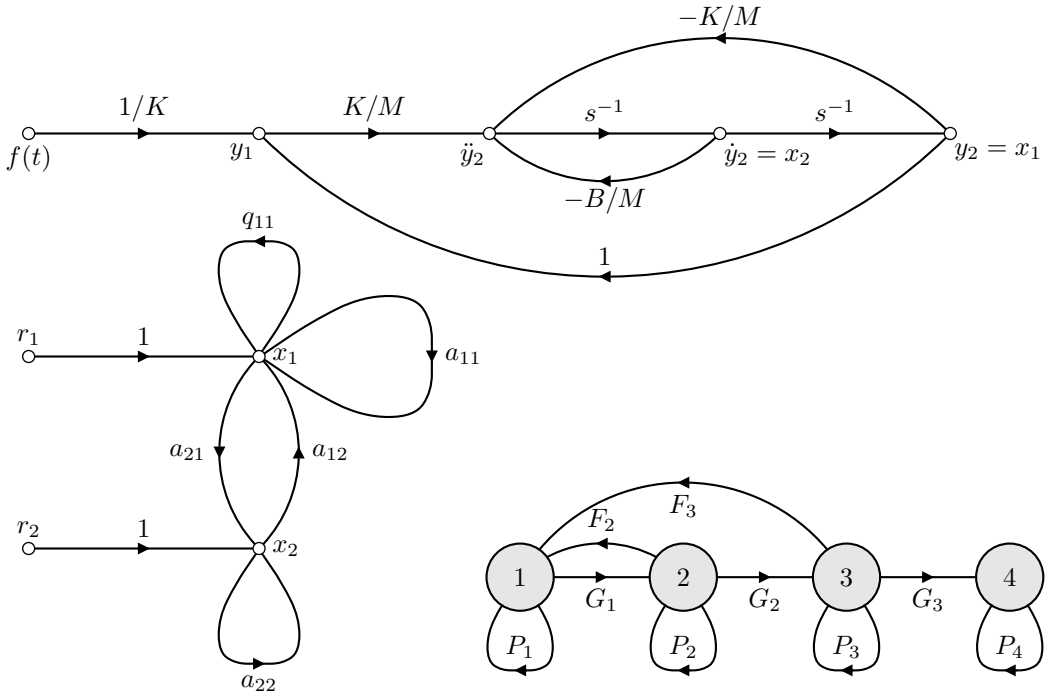


Figure 42: Signal-flow graphs [sfg.m4].

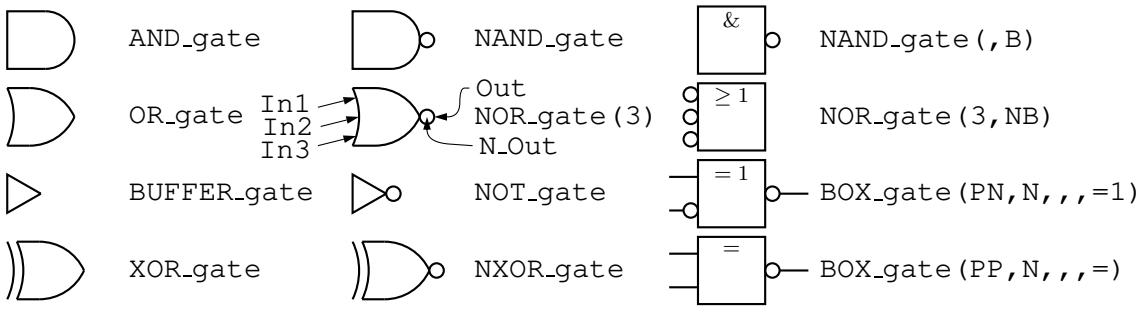


Figure 43: Basic logic gates [Logic.m4].

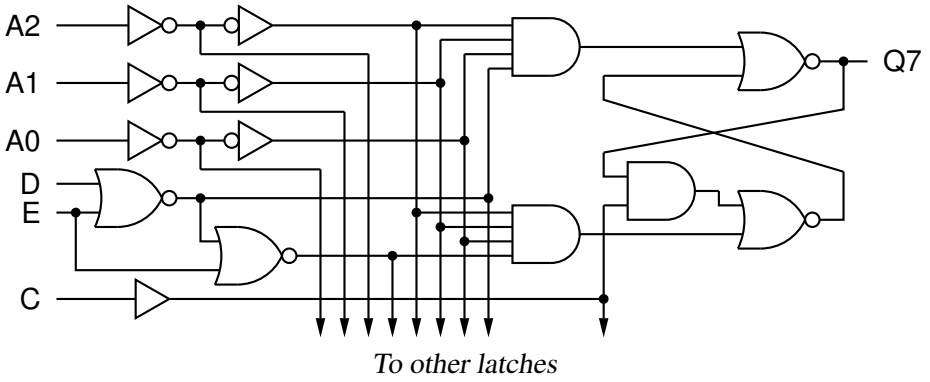


Figure 44: General-purpose latch: a small logic circuit [ex08.m4].

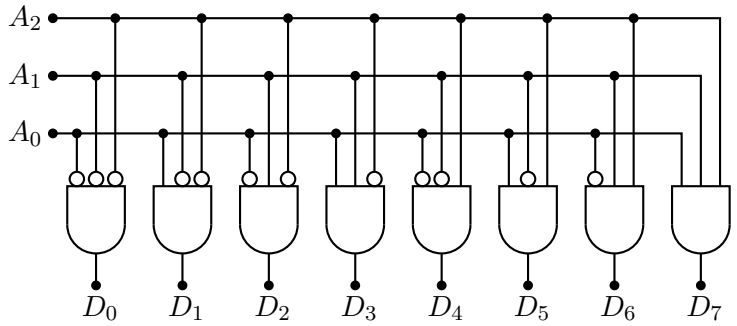


Figure 45: Decoder logic, constructed using the `for_` macro `[Decoder.m4]`.

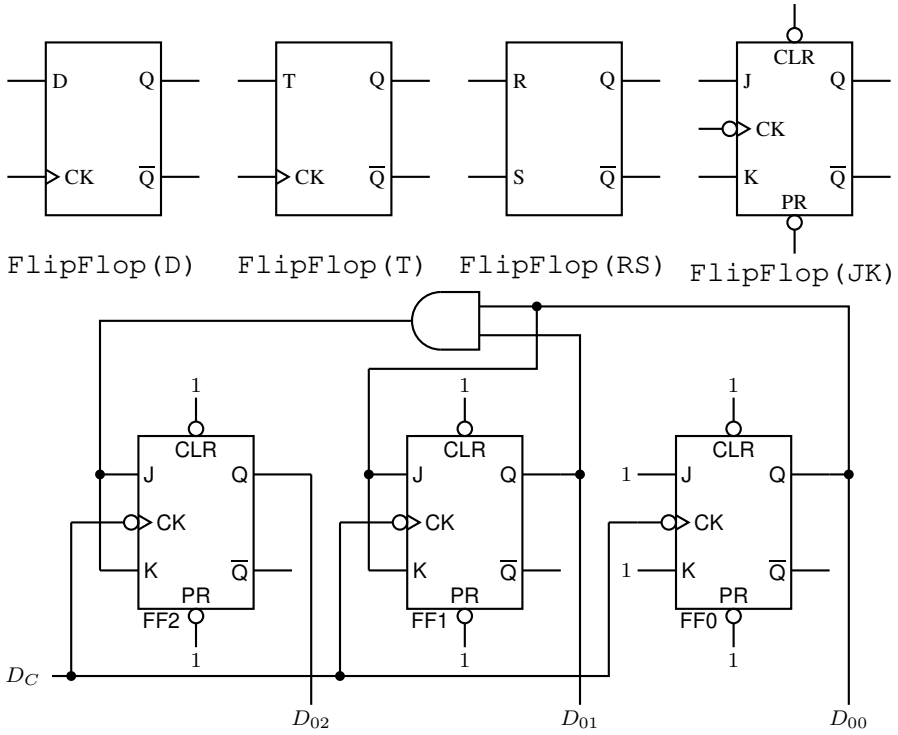


Figure 46: Some flip-flops `[ex21.m4]`.

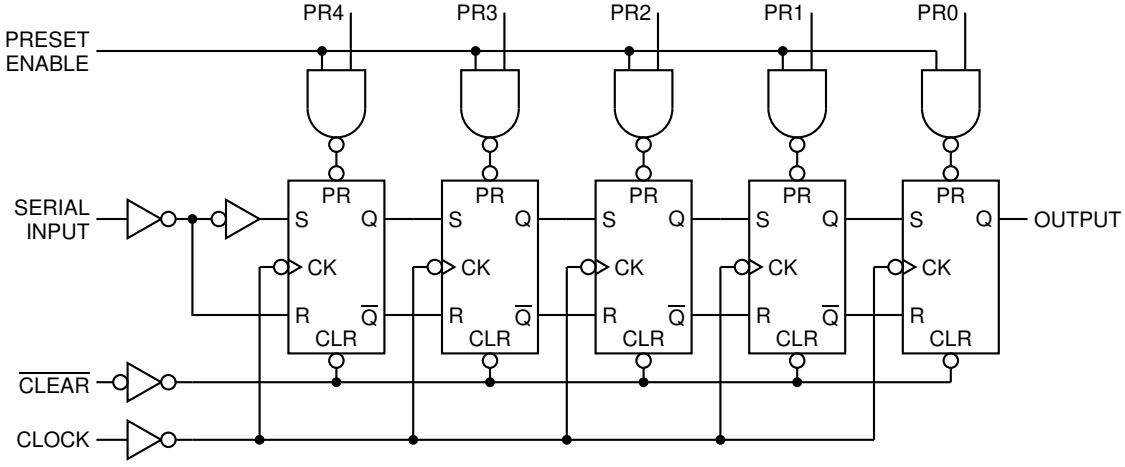
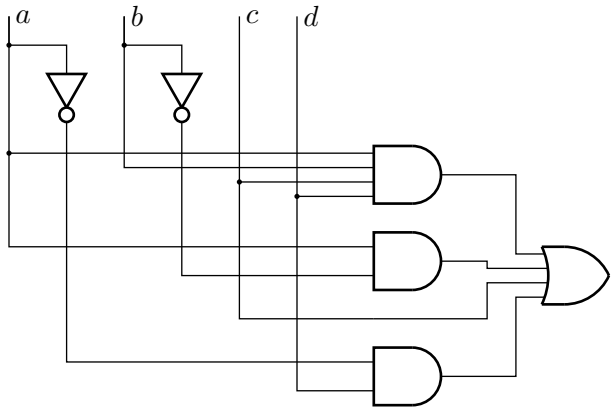
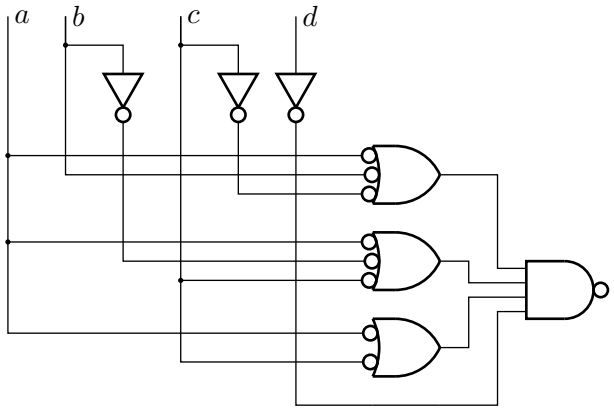


Figure 47: A 5-bit shift register drawn using a custom flip-flop `[ShiftR.m4]`.

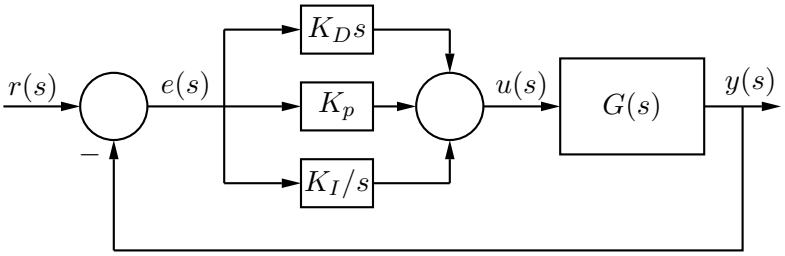


`CanLogic(AND,,OR,,abcd,a~b,c,~ad)`

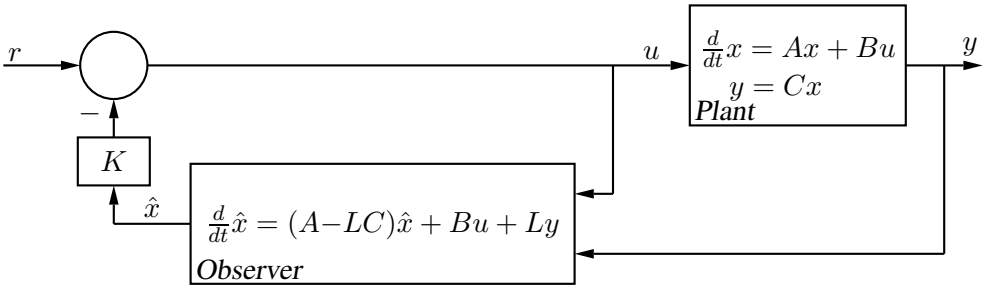


`CanLogic(OR,N,NAND,,ab~c,a~bc,ac,~d)`

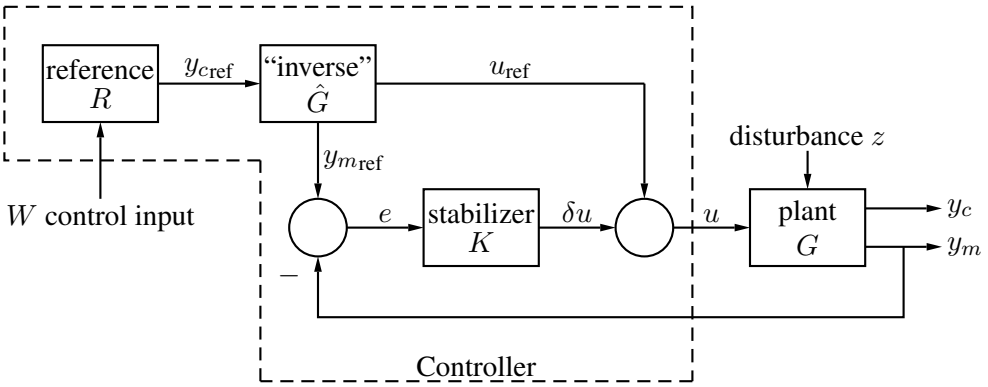
Figure 48: A way of automatically drawing two-layer logic diagrams `[CanLogic.m4]`.



(a) PID control



(b) Output feedback with a full-order observer



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

Figure 49: Control-system block diagrams that do not require m4 `[control.m4]`.

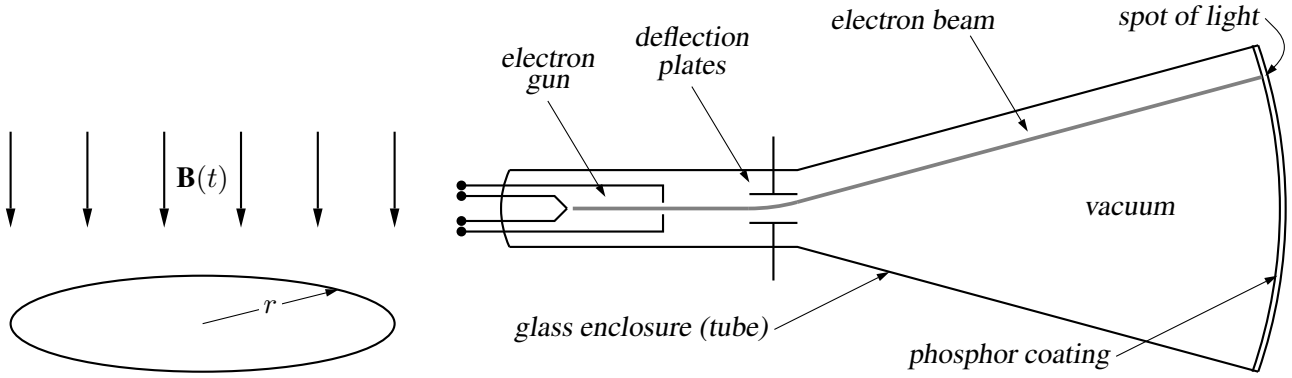


Figure 50: Line diagrams `[ex00.m4][ex07.m4]`.

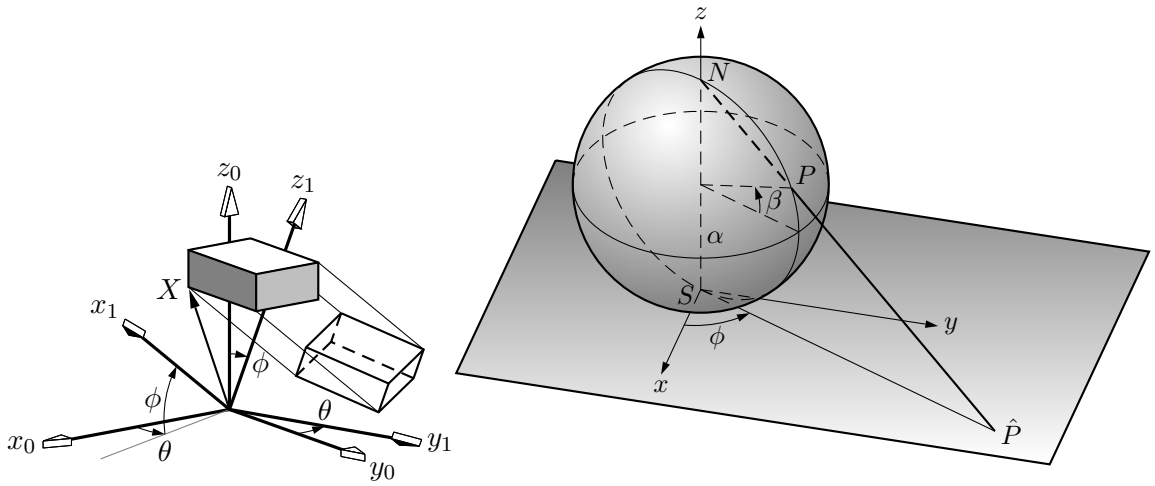


Figure 51: Test of `project` and other `lib3D` macros, showing the projection of a solid onto the y_1, z_1 plane by sighting along the x_1 axis. `[exp.m4][Globe.m4]`.

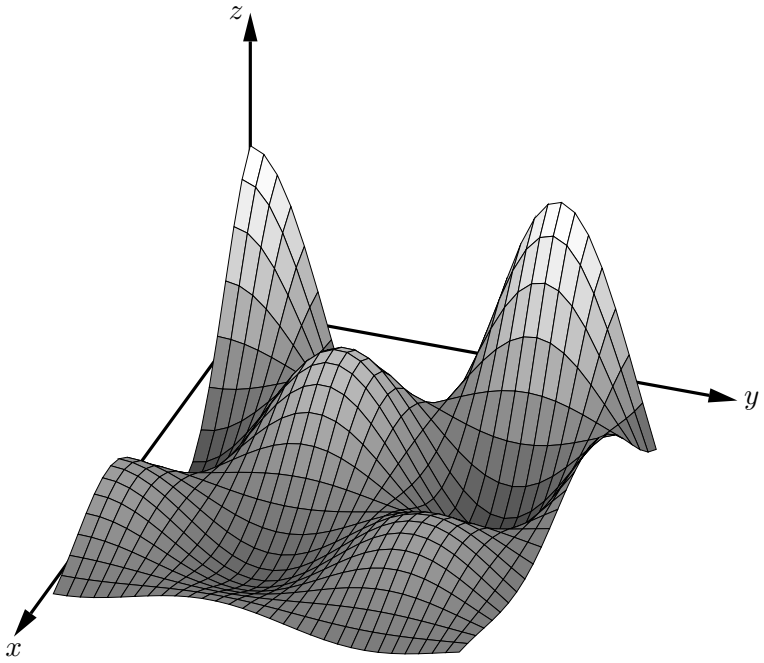
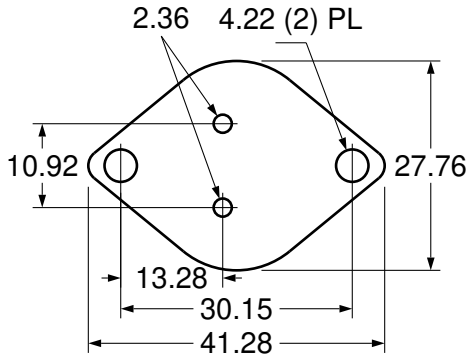
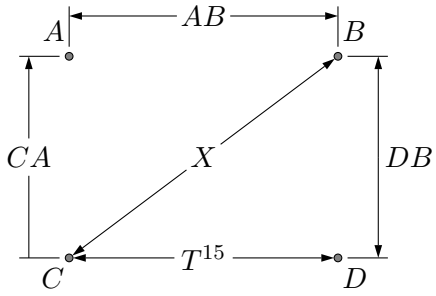
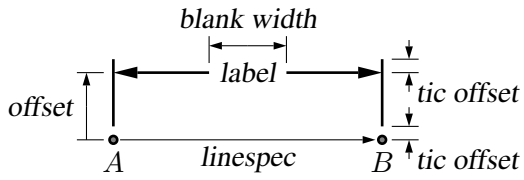


Figure 52: Plotting a surface using a gray scale `[graysurf.m4]`.






```
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)
```




```
dimension_(from A to B,0.5,\sl label,29bp__,0.1)
```

Figure 53: 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 [ex09.m4]`.

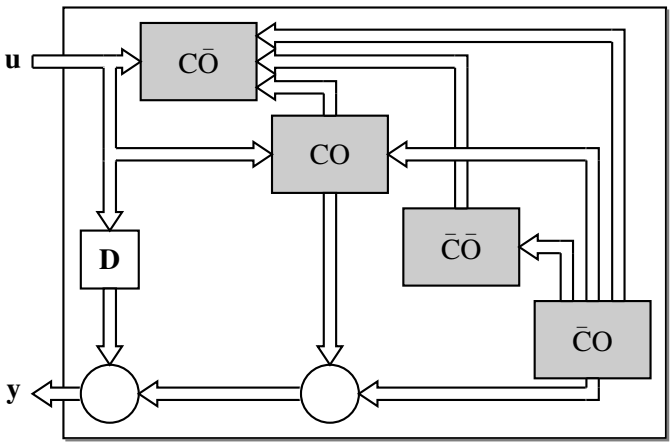


Figure 54: Use of `darrow [ex05.m4]`.

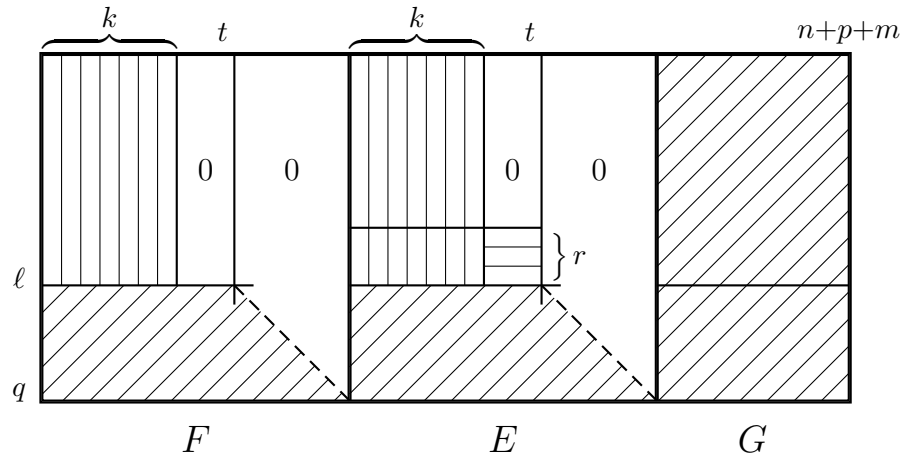


Figure 55: Crosshatching by `for` loops `[ex06.m4]`.

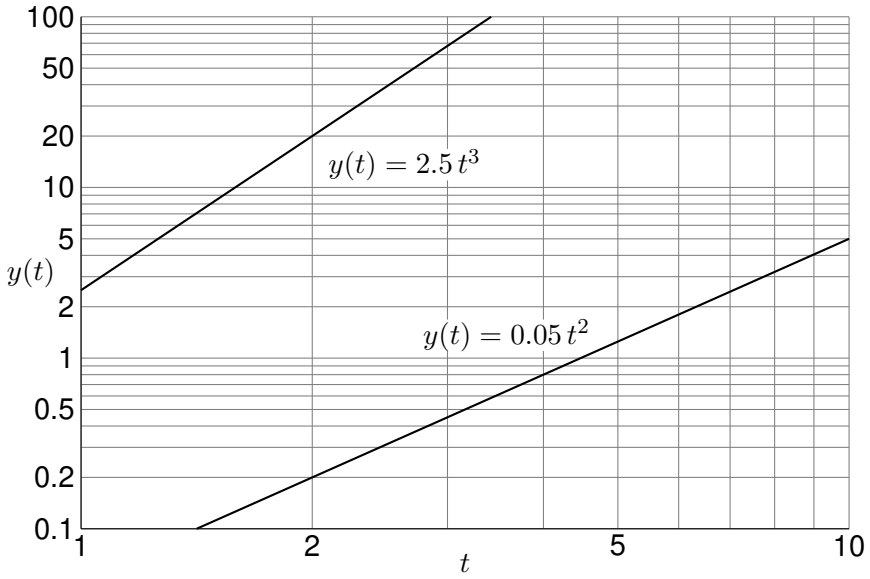


Figure 56: A graph drawn using the pic language `[Loglog.m4]`.

Figure 57: Conestoga Sailing Club (illustrating the filling of arbitrary shapes) `[csc.m4]`.

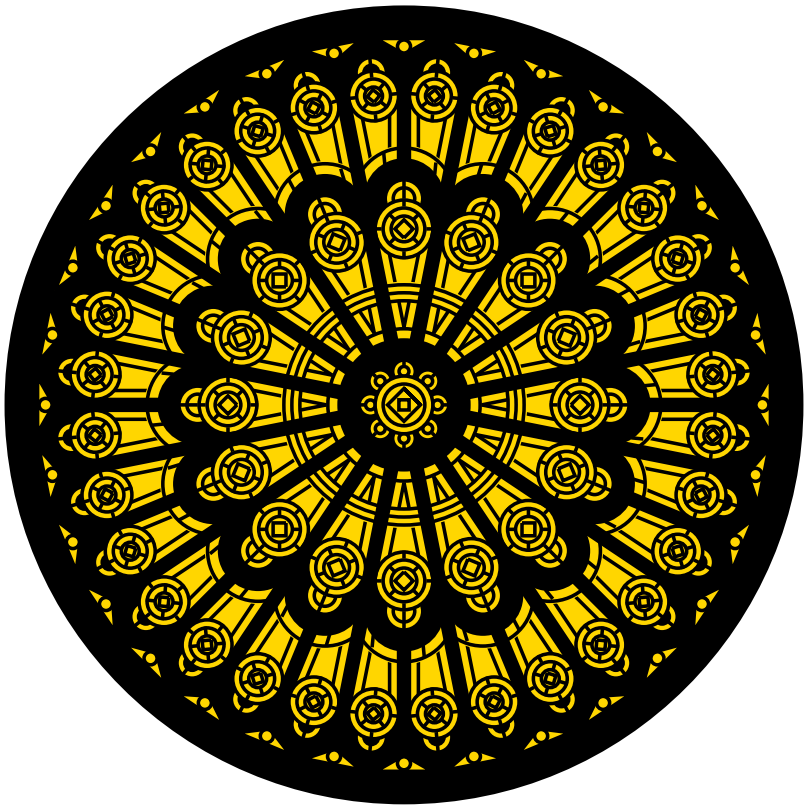


Figure 58: Redrawn from a detail of the set design for the musical *Dracula,* used for testing `dpic`. This diagram consumes much LaTeX main memory but can be produced as the raw postscript output of `dpic -r` since no text formatting is required `[rose.m4]`.

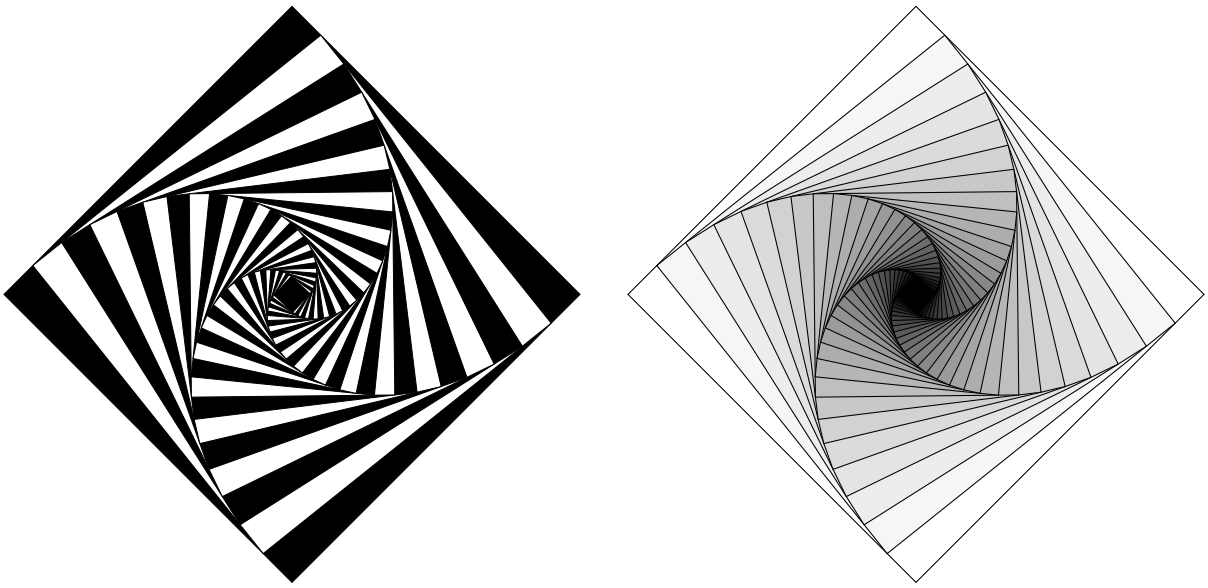


Figure 59: Variations on M. Goossens, S. Rahtz, and F. Mittelbach, *The LaTeX Graphics Companion,* Addison-Wesley 1997, pp. 57-58 `[diamond.m4]`.

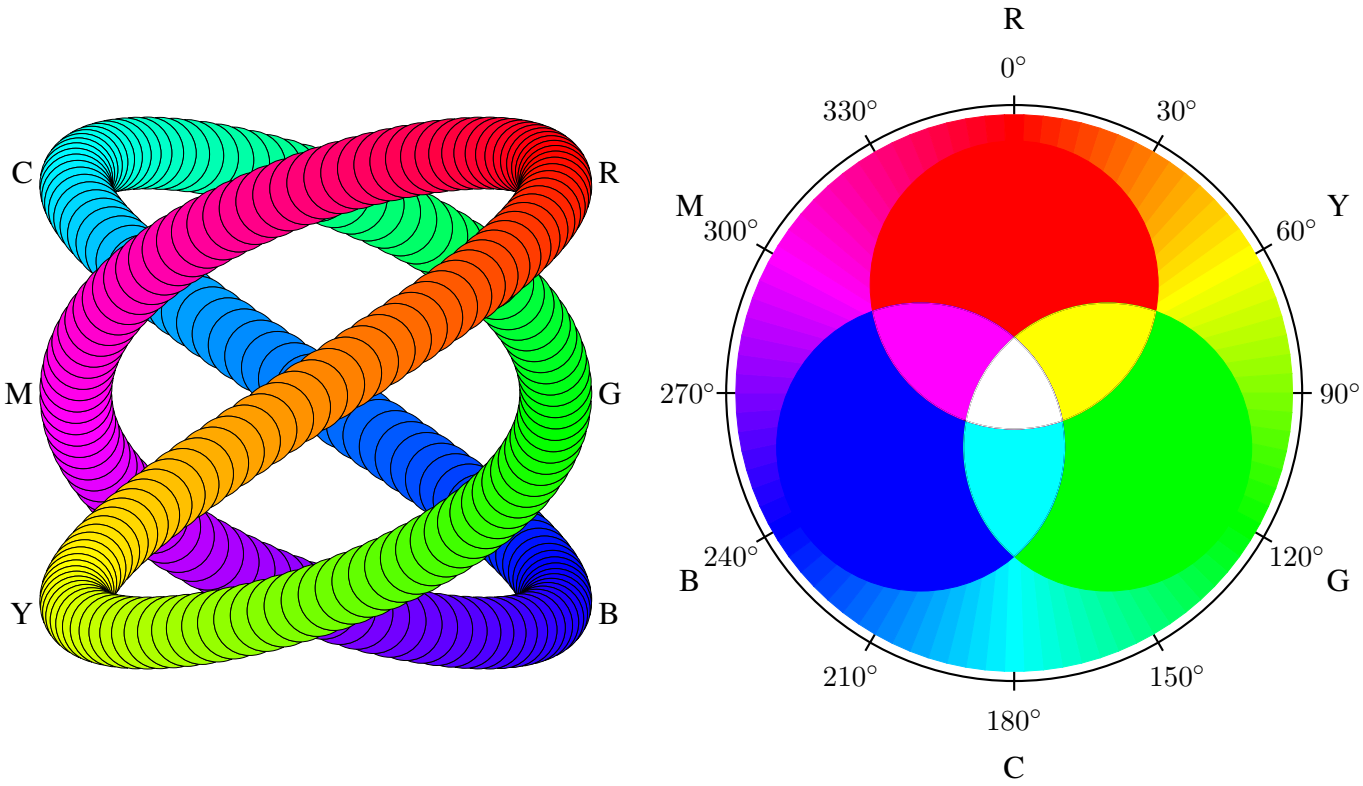


Figure 60: An exercise in calculating RGB colours `[worm.m4]`.

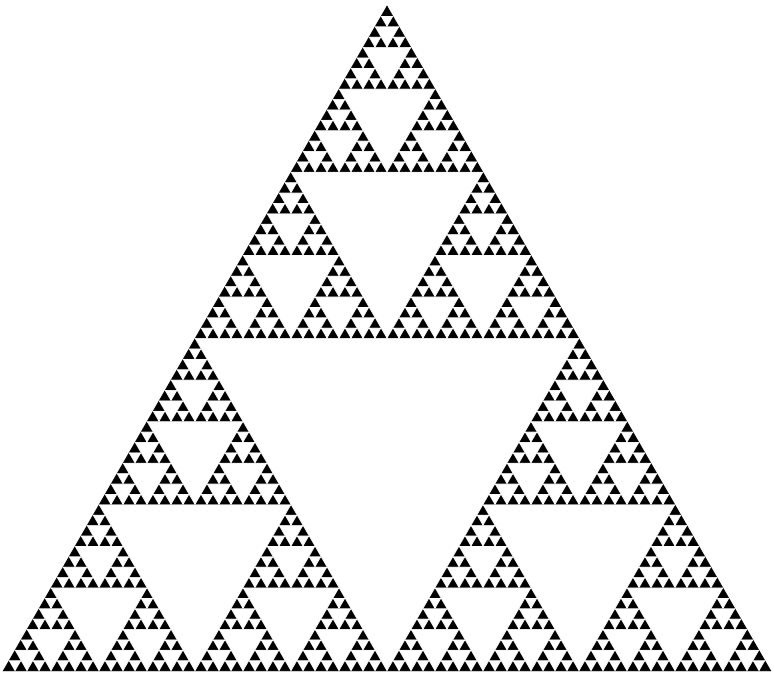


Figure 61: The Sierpinski triangle: a recursive test of pic macros `[Sierpinski.m4]`.

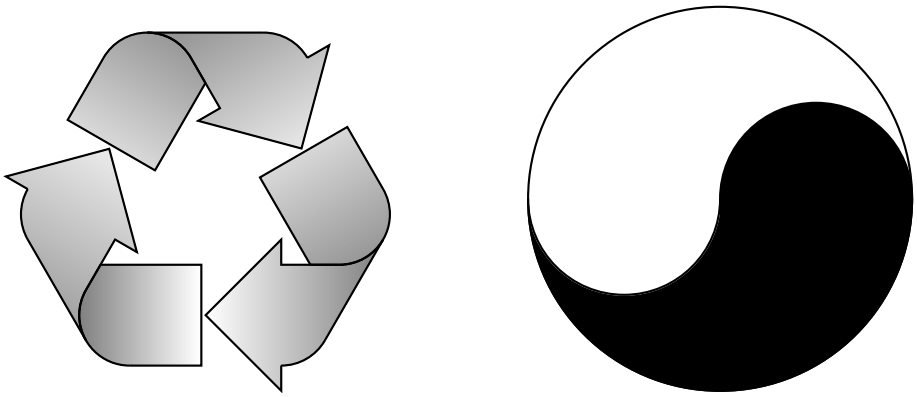


Figure 62: Modest repetition and partial fill `[recycle.m4][yinyang.m4]`.

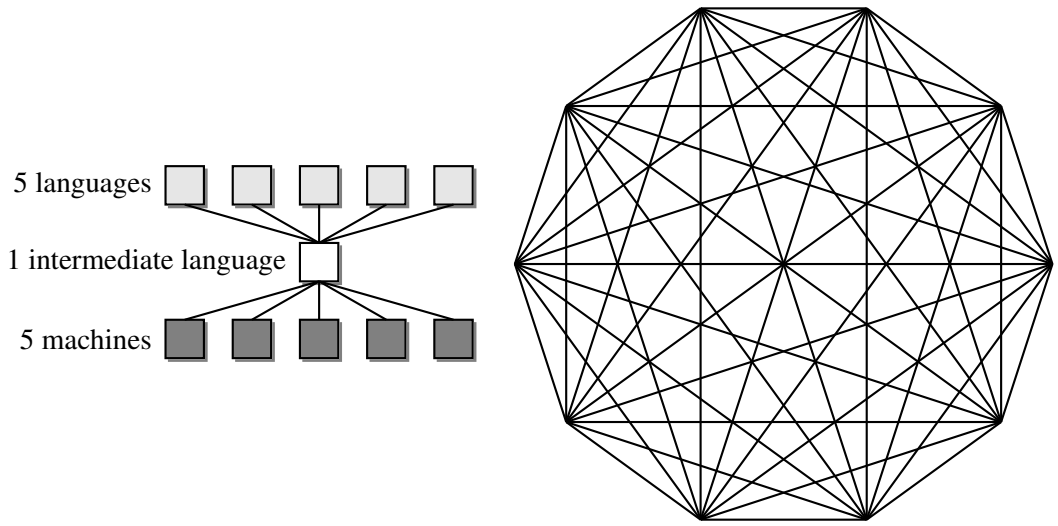


Figure 63: Simple diagrams that are easily drawn by looping `[ex15.m4]`.

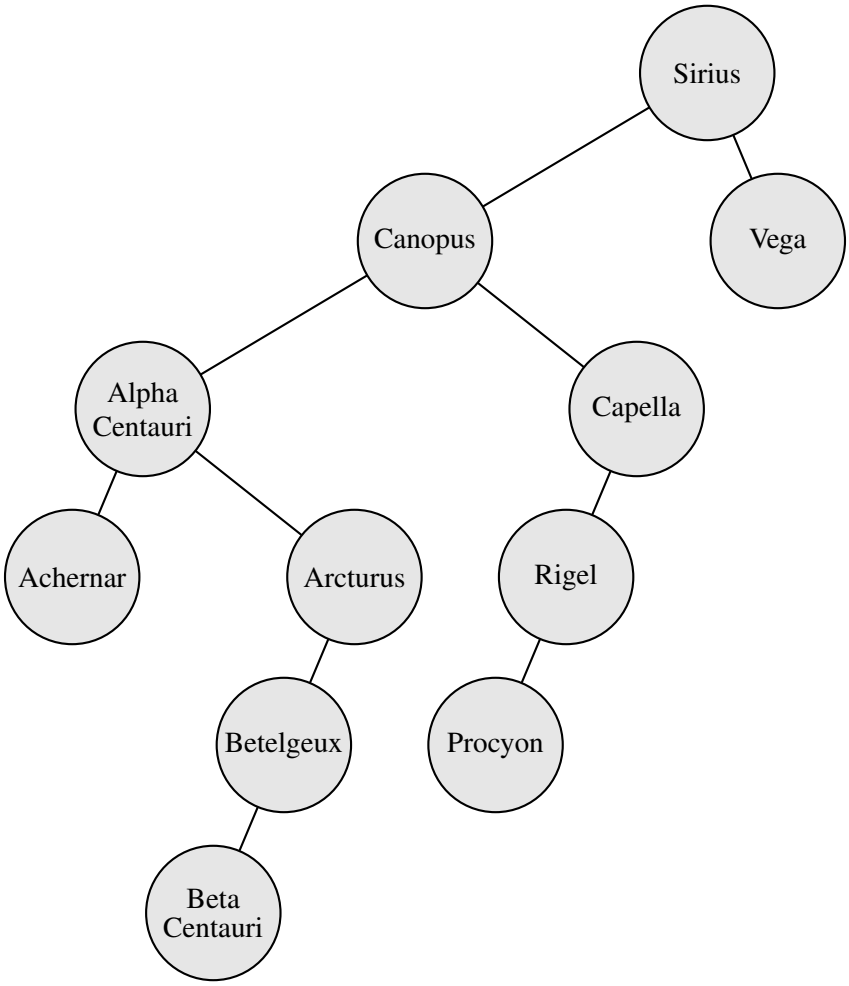


Figure 64: A binary tree `[Btree.m4]`.

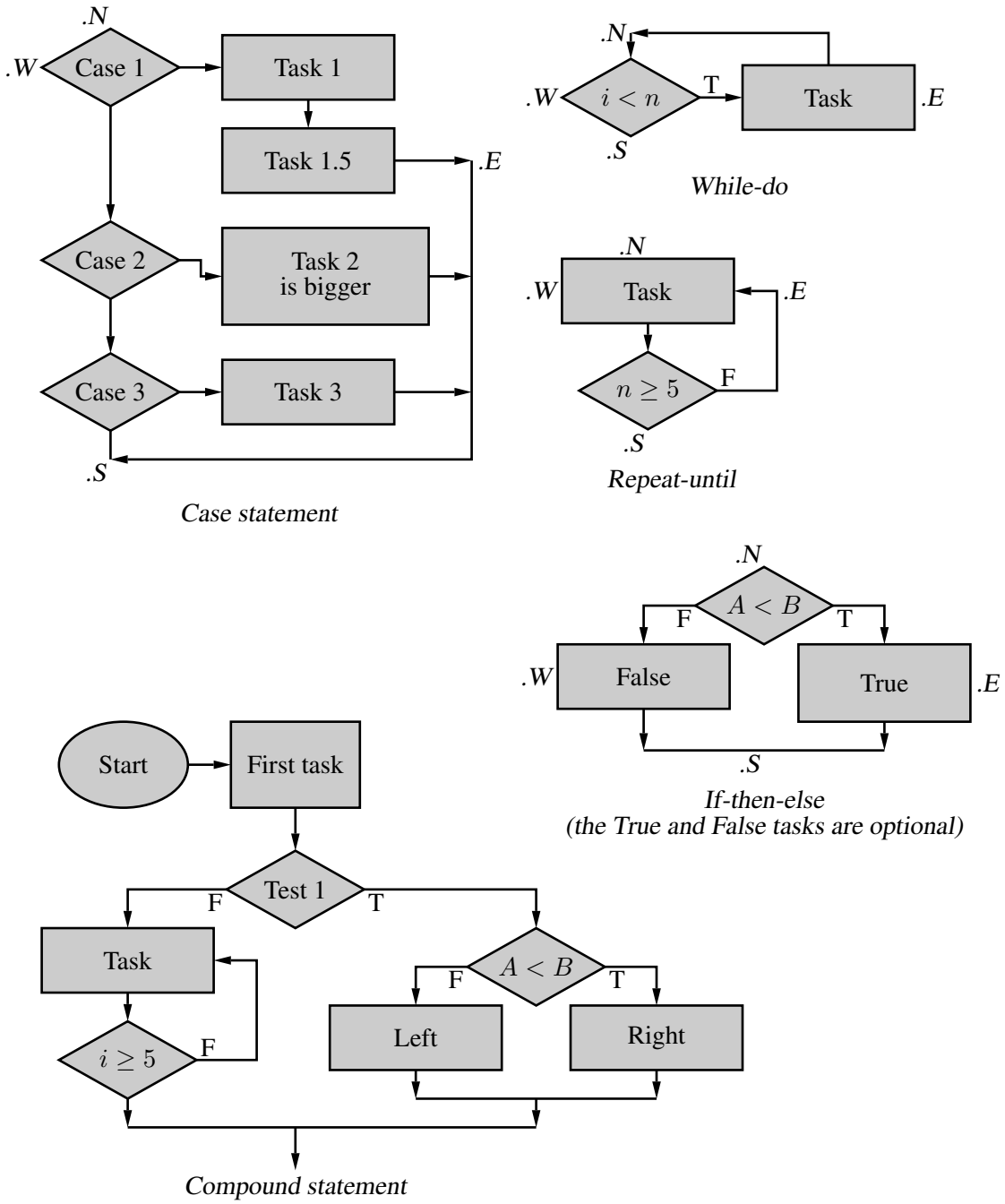


Figure 65: A flowchart sampler `[Flow.m4]`.

Figure 66: Overlaying a figure with line graphics `[Incleps.m4]`.