Fall 2003 BMI 226 / CS 426 Notes KKK-1

AUTOMATIC SYNTHESIS OF ELECTRICAL CIRCUITS USING DEVELOPMENTAL GENETIC PROGRAMMING

PART 3 — ACTIVE CIRCUIT EXAMPLES

DESIGN OF A 10 DB AMPLIFIER

- **Function set for construction-continuing subtrees**

F_{ccs} = {`R`, `C`, `QT0`, `QT1`, `QT2`, `QT3`, `QT4`,
`QT5`, `QT6`, `QT7`, `QT8`, `QT9`, `QT10`, `QT11`,
`SERIES`, `PARALLEL0`, `FLIP`, `NOP`,
`THGND`, `THPOS`, `THVIA0`, `THVIA1`,
`THVIA2`, `THVIA3`, `THVIA4`, `THVIA5`,
`THVIA6`, `THVIA7`}

- **Terminal set for construction-continuing subtrees**

T_{ccs} = {`END`, `CUT`}

- **Function set, for arithmetic-performing subtrees**

F_{aps} = {`+`, `-`}

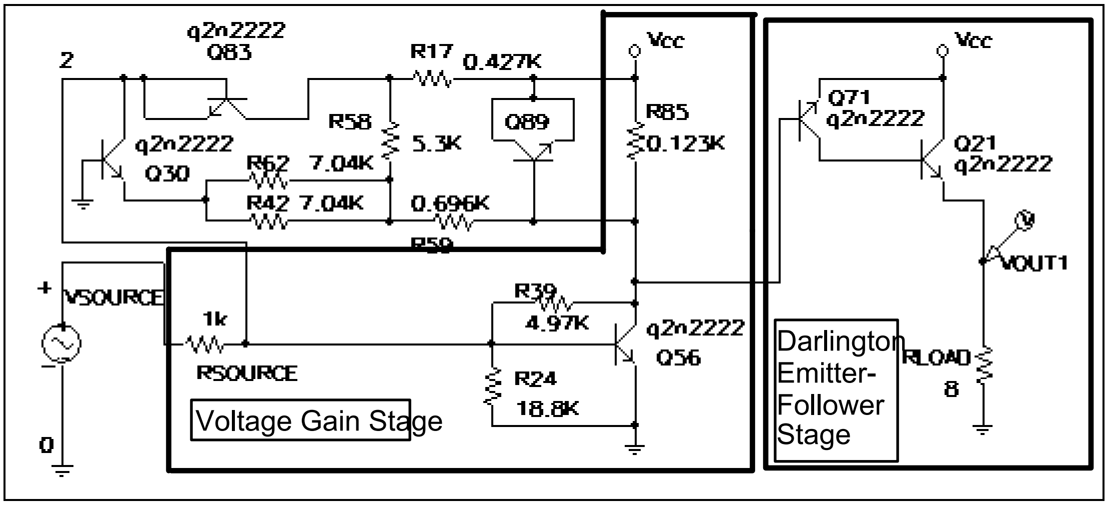
- **Terminal set for arithmetic-performing subtrees**

$T_{aps} = \{\Re\}$

FITNESS MEASURE FOR AN AMPLIFIER (10 DB PROBLEM)

- **Frequency domain**
- **The incoming AC signal source is 500 millivolts in amplitude**
- **Very simple**

REDRAWN BEST-OF-GENERATION GENETICALLY EVOLVED AMPLIFIER FROM GENERATION 45 SHOWING THE VOLTAGE GAIN STAGE AND DARLINGTON EMITTER FOLLOWER SECTION (10 DB PROBLEM)



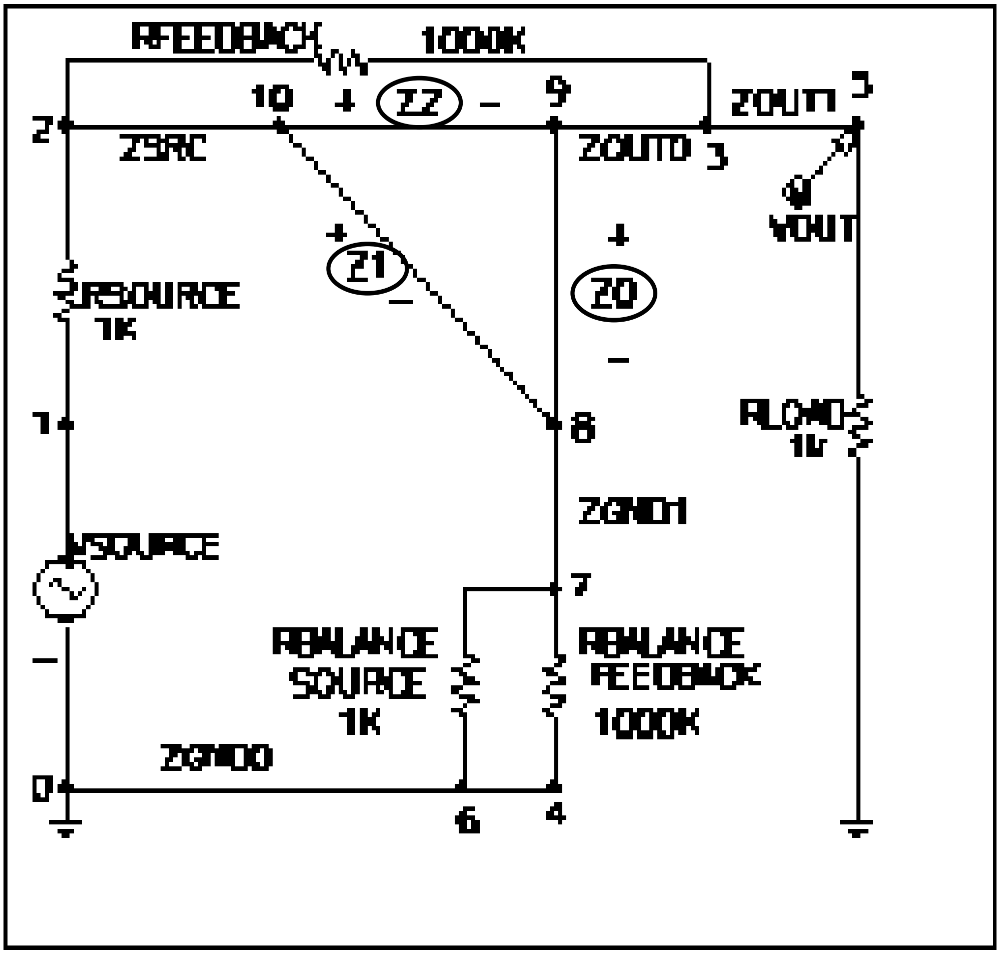
Darlington — U.S. Patent 2,663,806

TWELVE INSTANCES IN *GENETIC PROGRAMMING III* BOOK (1999) WHERE GENETIC PROGRAMMING APPEARS TO HAVE INFRINGED DARLINGTON'S PATENT (2,663,806)

Figure	Circuit	Transistors	Patent claim
45.16	96 dB amplifier	Q5 and Q25	1
45.16	96 dB amplifier	Q53, Q32	3
47.6	Squaring computational	Q101, Q119	1
47.6	Squaring computational	Q29, Q88	4
47.10	Cubing computational	Q27, Q46	3
47.10	Cubing computational	Q46, Q35	3
47.11	Cubing computational	Q35, Q49	3
47.12	Square root computational	Q120, Q155	2
47.15	Cube root computational	QNC19, QNC24	2
47.16	Cube root computational	QNC73, QNC74	1
47.16	Cube root computational	QNC74, QNC48	2
47.17	Logarithmic computational	Q22, Q66	4

FEEDBACK EMBRYO WITH THREE WRITING HEADS (60 DB AMPLIFIER)

(60 DB = 1000-TO-1)



ARCHITECTURE OF CIRCUIT-CONSTRUCTING PROGRAM TREE (60 DB AMPLIFIER)

- **The circuit-constructing program tree has three result-producing branches (called `RPB0`, `RPB1`, and `RPB2`).**

- **The circuit-constructing program tree also has two function-defining branches, each consisting of one-argument automatically defined function (called `ADF0` and `ADF1`).**

- **Each program tree has a total of five branches (i.e., two function-defining branches and three result-producing branches) joined by a connective `LIST` function.**

FUNCTION AND TERMINAL SETS OF RESULT PRODUCING BRANCHES (60 DB AMPLIFIER)

- **For the three result-producing branches, the function set, $F_{ccs\text{-}rpb}$, for each construction-continuing subtree is**

$F_{ccs\text{-}rpb}$ = {`ADF0`, `ADF1`, `R`, `C`, `SERIES`, `PARALLEL0`, `PARALLEL1`, `FLIP`, `NOP`, `NEW_T_GND_0`, `NEW_T_GND_1`, `NEW_T_POS_0`, `NEW_T_POS_1`, `NEW_T_NEG_0`, `NEW_T_NEG_1`, `PAIR_CONNECT_0`, `PAIR_CONNECT_1`, `Q_D_NPN,  Q_D_PNP`, `Q_3_NPN0`, ..., `Q_3_NPN11`, `Q_3_PNP0`, ..., `Q_3_PNP11`, `Q_POS_COLL_NPN`, `Q_GND_EMIT_NPN`, `Q_NEG_EMIT_NPN`, `Q_GND_EMIT_PNP`, `Q_POS_EMIT_PNP`, `Q_NEG_COLL_PNP`}

- **For the three result-producing branches, the terminal set, $T_{ccs\text{-}rpb}$, for each construction-continuing subtree consists of**

$T_{ccs\text{-}rpb}$ = {`END`, `SAFE_CUT`}.

FUNCTION AND TERMINAL SETS FOR FUNCTION-DEFINING BRANCHES (60 DB AMPLIFIER)

- **For the function-defining branches (automatically defined functions), the function set, $F_{\text{ccs-adf}}$, for each construction-continuing subtree is**

$F_{\text{ccs-adf}} = F_{\text{ccs-rpb}} - \{\texttt{ADF0}, \texttt{ADF1}\}$.

- **The terminal sets are identical for both function-defining branches (automatically defined functions) of the program trees for this problem. The function sets are identical for both function-defining branches (automatically defined functions).**

- **For the function-defining branches, the terminal set, $T_{\text{ccs-adf}}$, for each construction-continuing subtree is**

$T_{\text{ccs-adf}} = T_{\text{ccs-adf}} \approx \{\texttt{ARG0}\}$.

FUNCTION AND TERMINAL SETS FOR ARITHMETIC-PERFORMING SUBTREES (FOUND IN BOTH RBPs AND ADFs) (60 DB AMPLIFIER)

- **The terminal set, T_{aps}, for each arithmetic-performing subtree consists of**

Taps = $\{\Re\}$,

where $\Re$ represents floating-point random constants from –1.0 to +1.0.

- **The function set, F_{aps}, for each arithmetic-performing subtree is,**

Faps = {+, -}.

FITNESS MEASURE (60 DB AMPLIFIER)

- **Fitness is the sum of the amplification penalty, the bias penalty, and the two non-linearity penalties.**

- **First, the overall amplification factor of the circuit is measured using the overall value for gain of the circuit as measured by the slope of the straight line between the output for –10 millivolts and the output for +10 millivolts (i.e., between the outputs for the endpoints of the DC sweep).**

  - If the amplification factor is less than the maximum possible amplification (1,000-to-1 for this problem), the penalty is equal to the numerical difference between the maximum possible gain and the actual gain. For example, if the gain were 100-to-1, there would be a detrimental contribution of 900 to the fitness measure.

FITNESS MEASURE (60 DB AMPLIFIER) – CONTINUED

- **Second, the linearity is measured by the deviation between the slope of each of two shorter lines and the overall amplification factor of the circuit. The first shorter line segment connects the output value associated with an input of –10 mv and the output value associated with an input of –5 mv.**

The second shorter line segment connects the output value for +5 mv and the output value for +10 mv. Each of these two shorter line segments contributes to the fitness.

The detrimental contribution to fitness of each shorter line segment is equal to the weighted absolute value of the difference between the slope of shorter line segment and the overall amplification factor of the circuit.

FITNESS MEASURE (60 DB AMPLIFIER) – CONTINUED

- **Third, the bias is computed using the DC output associated with a DC input of 0 volts.**

There is a penalty equal to the bias times a weight (which, for this problem, is 0.1).

- **Unsimulatable programs = 10^8 penalty**

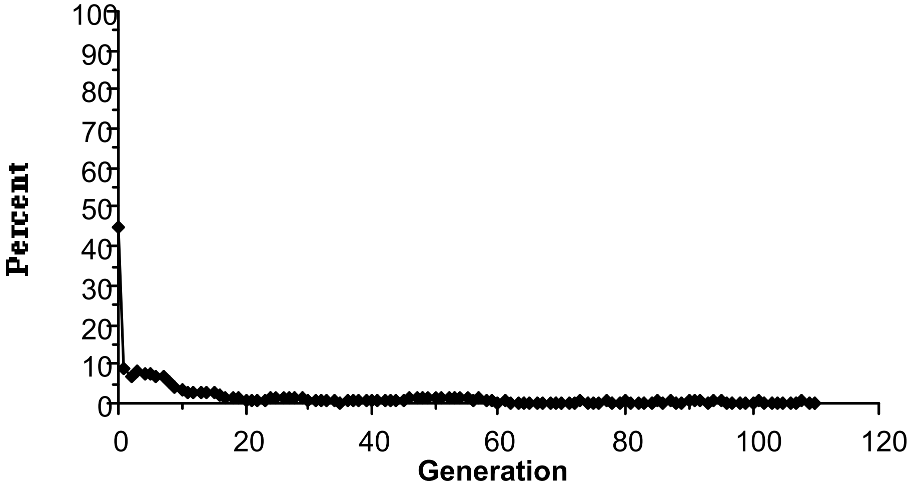
CONTROL PARAMETERS (60 DB AMPLIFIER)

- **Population size, *M*, of 640,000**
- **Maximum number of generations, *G*, is set to be meaninglessly large**
- **Maximum of H_{rpb} = 300 points (functions and terminals) for each result-producing branch**
- **Maximum of H_{adf} = 300 points (functions and terminals) for each function-defining branch**
- **For each generation**
  - 10% reproductions
  - 1% mutations
  - 89% crossovers
- **Secondary parameters are default values in Koza 1994 (appendix D)**

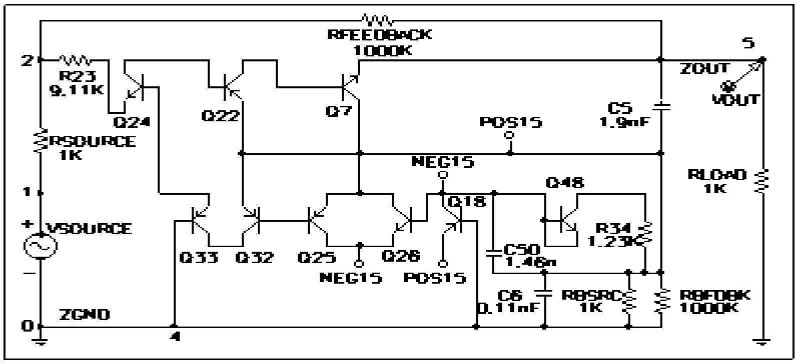
TERMINATION CRITERION AND RESULTS DESIGNATION (60 DB AMPLIFIER)

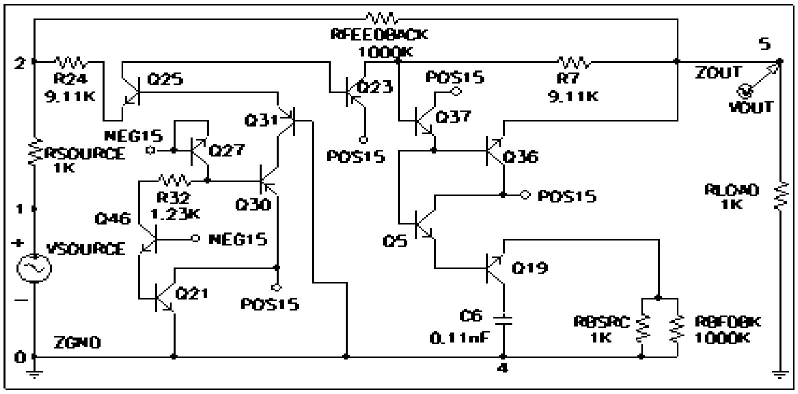
- **Manual intervention in lieu of pre-established termination criterion**
- **Best-so-far individual is designated as the result of the run**

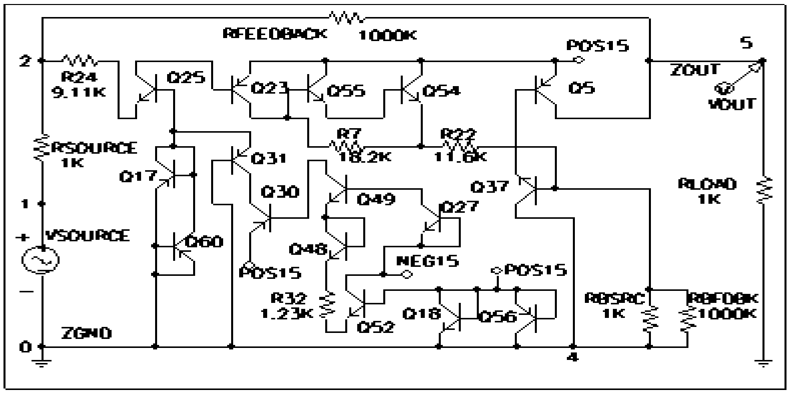
PERCENTAGE OF UNSIMULATABLE PROGRAMS (60 DB)

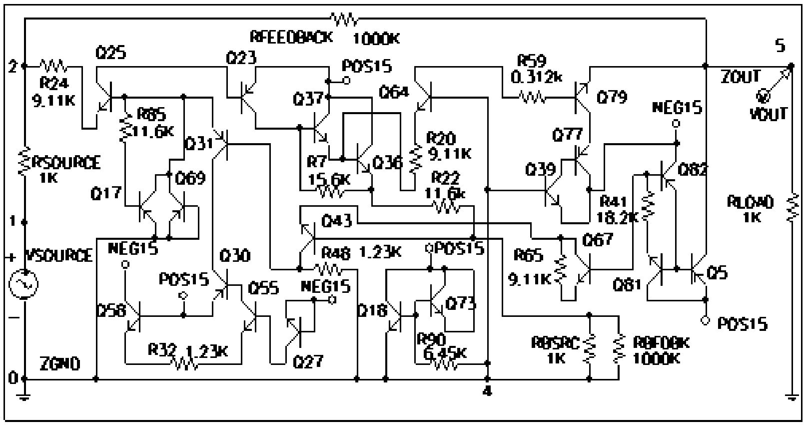


BEST OF GENS 0, 19, 49,109 (60 DB)

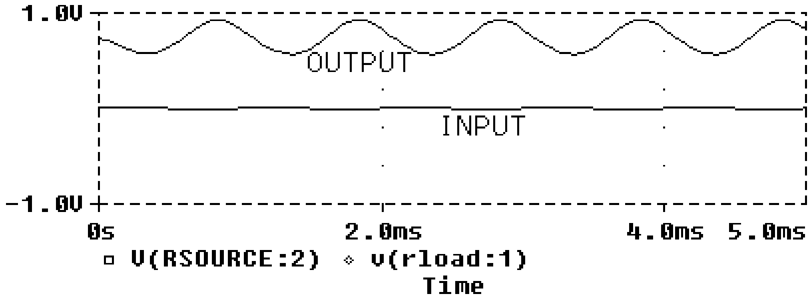


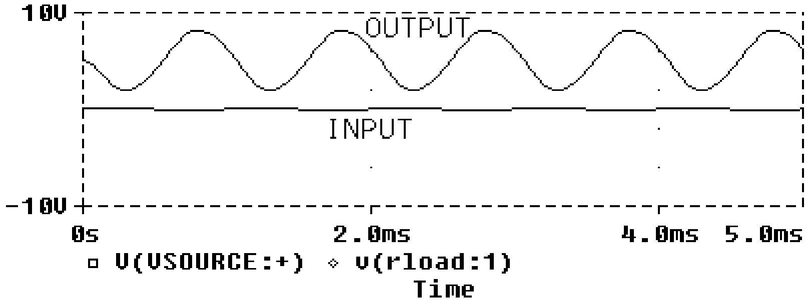


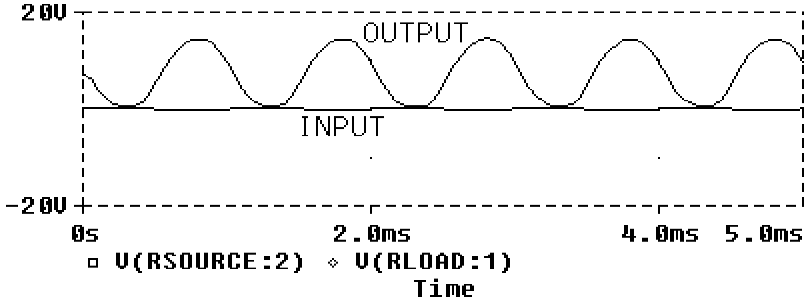


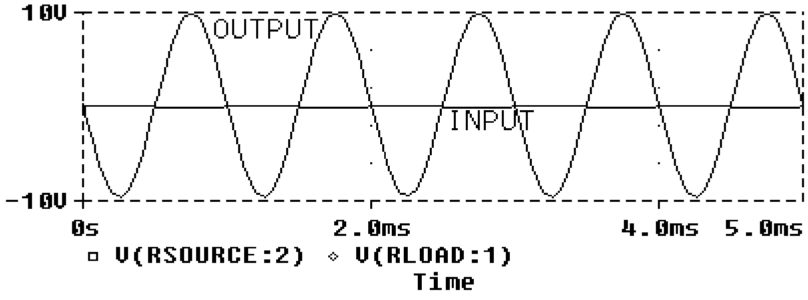


TIME DOMAIN – BEST OF GENS 0, 19, 49,109 (60 DB)

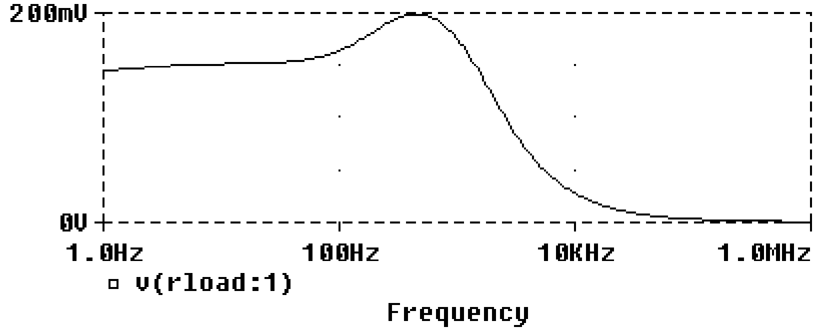


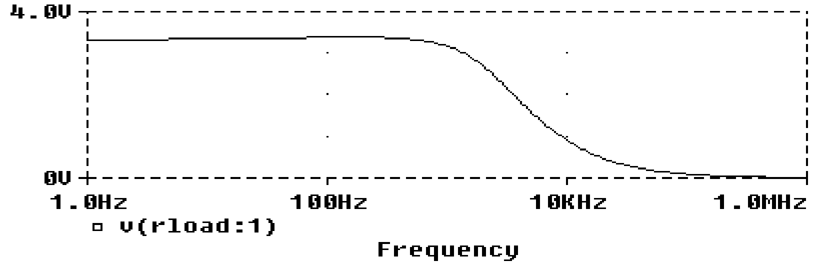


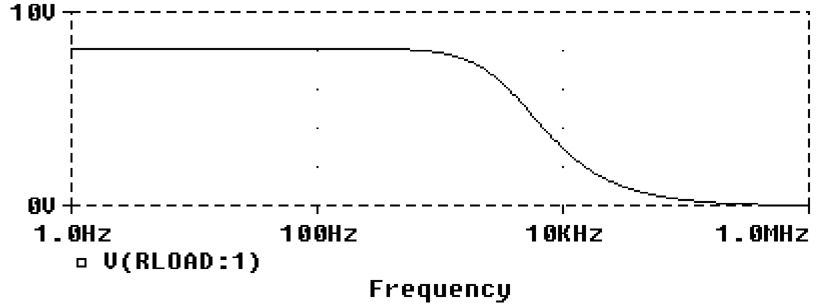


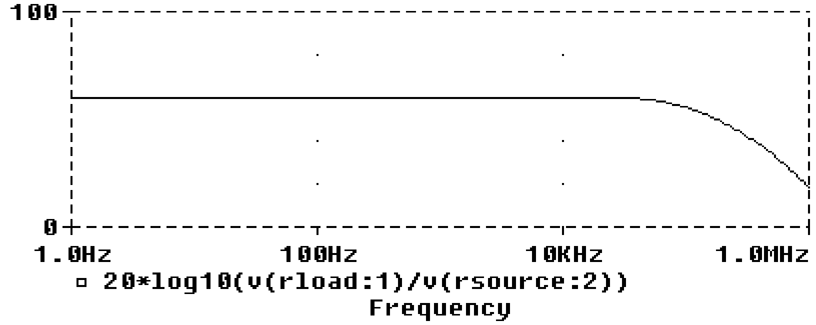


AC SWEEPS – BEST OF GENS 0, 19, 49,109 (60 DB)

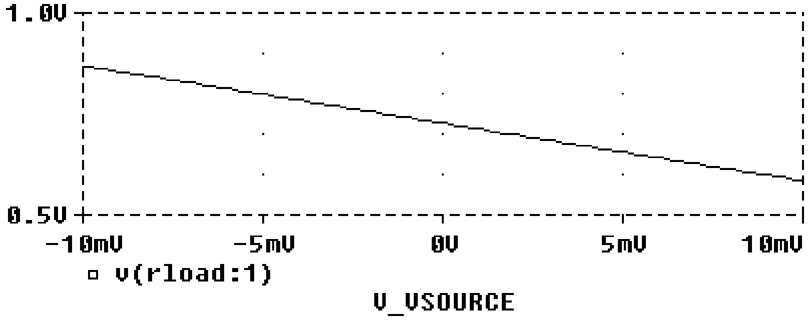


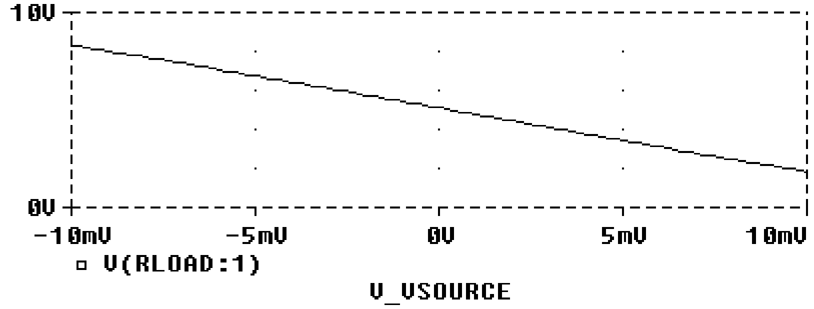


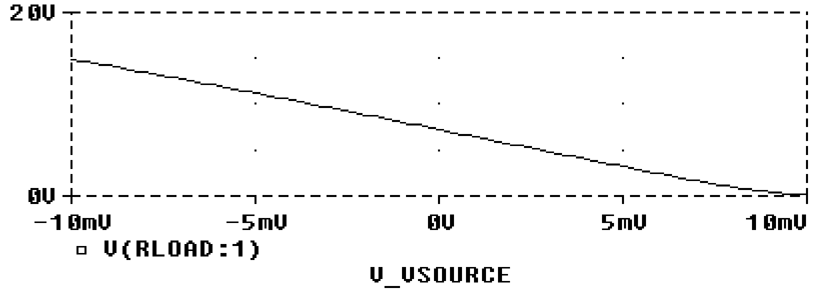


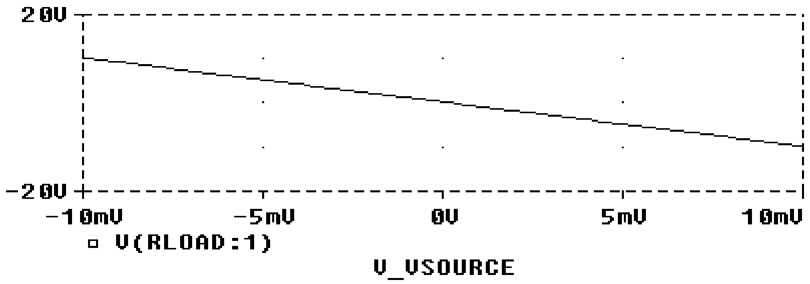


DC SWEEP – BEST OF GENS 0, 19, 49,109 (60 DB)



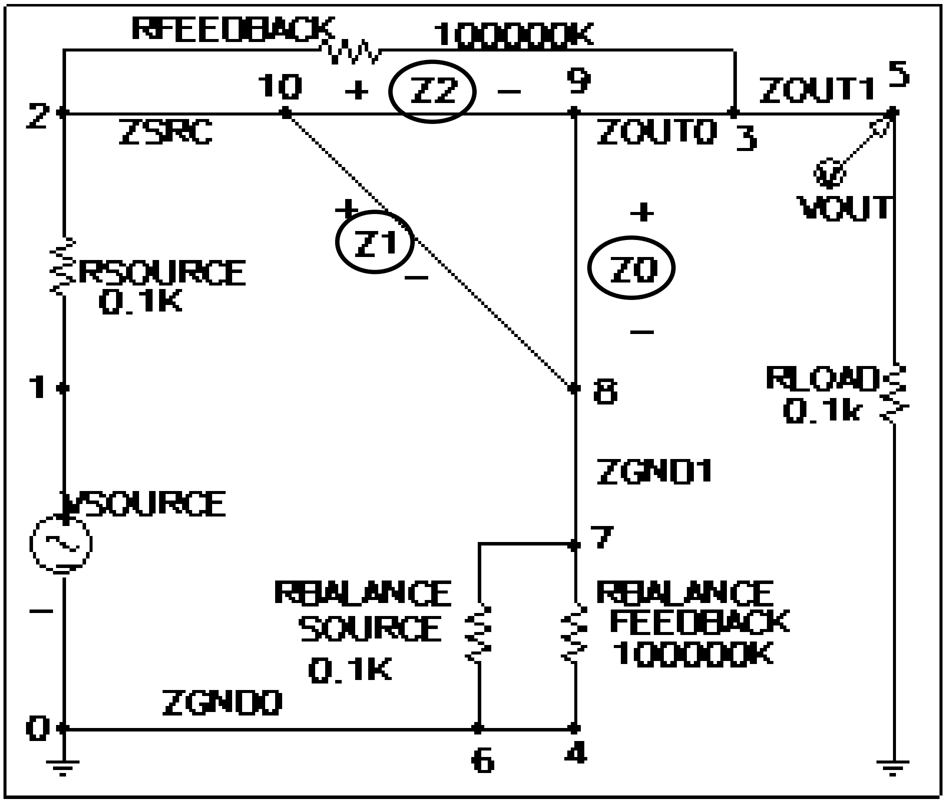






ONE-INPUT, ONE-OUTPUT FEEDBACK EMBRYO WITH THREE WRITING HEADS (FOR 96 DB AMPLIFIER)

THESE FEEDBACK RESISTORS LIMIT AMPLIFICATION TO 120 DB



ARCHITECTURE OF CIRCUIT-CONSTRUCTING PROGRAM TREE (FOR 96 DB AMPLIFIER)

- **The circuit-constructing program tree has three result-producing branches (called `RPB0`, `RPB1`, and `RPB2`).**
- **The number of automatically defined functions, if any, will emerge as a consequence of the evolutionary process using the architecture-altering operations.**
- **Each program in the initial population of programs has a uniform architecture with no automatically defined functions (i.e., three result-producing branches).**
- **A connective `LIST` function joins the three result-producing branches) to whatever function-defining branches, if any, are present.**

ARCHITECTURE-ALTERING OPERATIONS

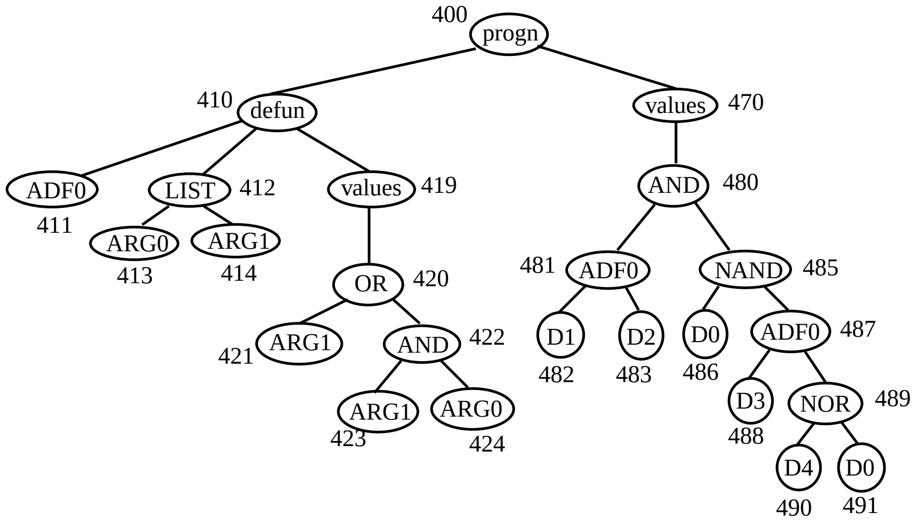
- **Branch duplication**
- **Argument duplication**

- **Branch deletion**
- **Argument deletion**

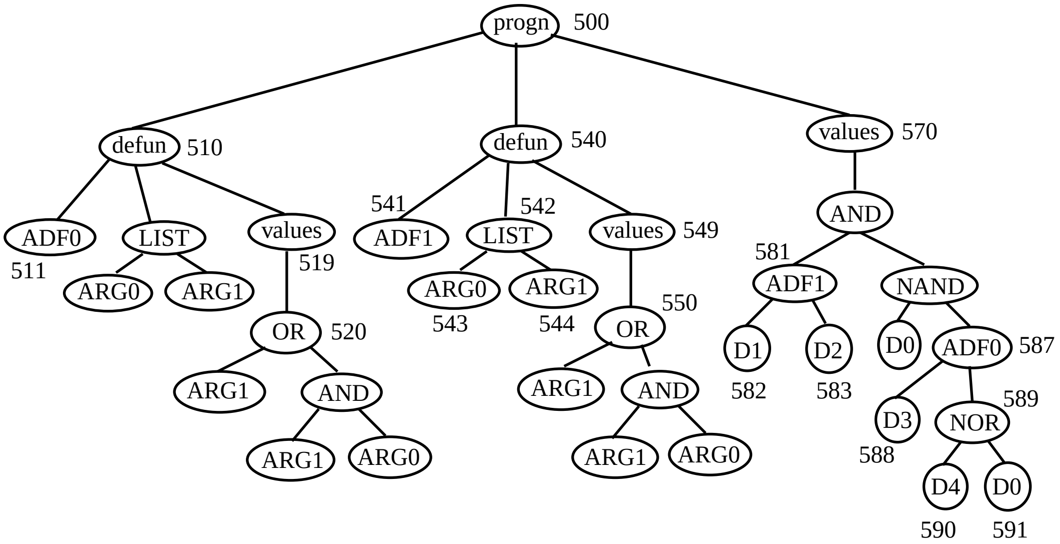
- **Branch creation**
- **Argument creation**

ARCHITECTURE-ALTERING OPERATIONS

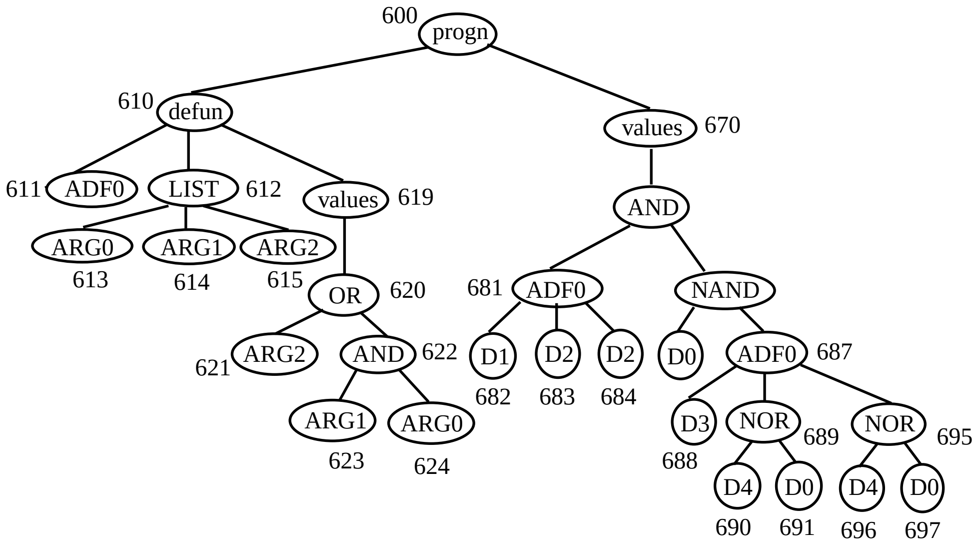
PROGRAM WITH 1 TWO-ARGUMENT AUTOMATICALLY DEFINED FUNCTION (`ADF0`) AND 1 RESULT-PRODUCING BRANCH – ARGUMENT MAP OF {2}



PROGRAM WITH ARGUMENT MAP OF {2, 2} CREATED USING THE OPERATION OF BRANCH DUPLICATION



PROGRAM WITH ARGUMENT MAP OF {3} CREATED USING THE OPERATION OF ARGUMENT DUPLICATION



FUNCTION AND TERMINAL SETS OF RESULT PRODUCING BRANCHES (FOR 96 DB AMPLIFIER)

- **The initial function set, Fccs-initial, for each construction-continuing subtree is**

$F_{ccs\text{-}rpb}$ = {`R`, `C`, `SERIES`, `PARALLEL0`, `PARALLEL1`, `FLIP`, `NOP`, `NEW_T_GND_0`, `NEW_T_GND_1`, `NEW_T_POS_0`, `NEW_T_POS_1`, `NEW_T_NEG_0`, `NEW_T_NEG_1`, `PAIR_CONNECT_0`, `PAIR_CONNECT_1`, `Q_D_NPN`, `Q_D_PNP`, `Q_3_NPN0`, ..., `Q_3_NPN11`, `Q_3_PNP0`, ..., `Q_3_PNP11`, `Q_POS_COLL_NPN`, `Q_GND_EMIT_NPN`, `Q_NEG_EMIT_NPN`, `Q_GND_EMIT_PNP`, `Q_POS_EMIT_PNP`, `Q_NEG_COLL_PNP`}

- **The initial terminal set, Tccs-initial, for each construction-continuing subtree is**

$T_{ccs\text{-}rpb}$ = {`END`, `SAFE_CUT`}.

FUNCTION AND TERMINAL SETS FOR FUNCTION-DEFINING BRANCHES (FOR 96 DB AMPLIFIER)

The set of potential new functions, $F_{potential}$, is

$F_{potential}$ = {`ADF0`, `ADF1`, `ADF2`, `ADF3`}.

The set of potential new terminals, $T_{potential}$, is

$T_{potential}$ = {`ARG0`}.

- **The architecture-altering operations change the function set, F_{ccs} for each construction-continuing subtree of all three result-producing branches and the function-defining branches, so**

$F_{ccs} = F_{ccs\text{-}initial} \approx F_{potential}$.

FUNCTION AND TERMINAL SETS FOR ARITHMETIC-PERFORMING SUBTREES (FOUND IN BOTH RBPs AND ADFs) (FOR 96 DB AMPLIFIER)

- **The terminal set, Taps, for each arithmetic-performing subtree consists of**

Taps = $\{\Re\}$,

where $\Re$ represents floating-point random constants from –1.0 to +1.0.

- **The function set, Faps, for each arithmetic-performing subtree is,**

Faps = {+, -}.

FITNESS MEASURE (96 DB AMPLIFIER)

- **An ideal inverting amplifier circuit would receive a DC input, invert it, and multiply it by the amplification factor.**
- **The starting point for evaluating the fitness of a circuit is its response to a DC input. A circuit is flawed to the extent that it does not achieve the desired amplification; to the extent that the output signal is not centered on 0 Volts (i.e., it has a bias); and to the extent that the DC response is not linear.**
- **The fitness measure is based on SPICE's DC sweep. The circuits were analyzed with a 5 point DC sweep ranging from –10 millvolts to +10 mV, with input points at –10 mV, –5 mV, 0 mV, +5 mV, and +10 mV.**
- **Fitness is the sum of**
  - the amplification penalty,
  - the bias penalty, and
  - the two non-linearity penalties.

FITNESS MEASURE (FOR 96 DB AMPLIFIER)

- **AMPLIFICATION PENALTY:** **The amplification factor of the circuit is measured by the slope of the straight line between the output for –10 mV and the output for +10 mV (i.e., between the outputs for the endpoints of the DC sweep). If the amplification factor is less than the maximum allowed by the feedback resistor (120 dB for this problem), there is a penalty equal to the shortfall in amplification.**

- **BIAS PENALTY:** **The bias is computed using the DC output associated with a DC input of 0 Volts. The penalty is equal to the bias times a weight. For this problem, a weight of 0.1 is used.**

FITNESS MEASURE (FOR 96 DB AMPLIFIER)

- **<u>TWO NON-LINEARITY PENALTIES</u>: The linearity is measured by the deviation between the slope of each of two line segments and the overall amplification factor of the circuit. The first line segment spans the output values associated with inputs of –10 mv through –5 mv. The second line segment spans the output values associated with inputs of +5 mv and through +10 mv. The penalty for each of these line segments is equal to the absolute value of the difference in slope between the respective line segment and amplification factor of the circuit.**

- **Unsimulatable programs = 10^8 penalty**

CONTROL PARAMETERS (FOR 96 DB AMPLIFIER)

- **Population size, *M*, of 640,000**

- **Maximum number of generations, *G*, is set to be meaninglessly large**

CONTROL PARAMETERS (96 DB AMPLIFIER) – CONTINUED

- **The percentage of operations on each generation after generation 5 was**
  - 86.5% one-offspring crossovers;
  - 10% reproductions;
  - 1% mutations;
  - 1% branch duplications;
  - 0.5% branch deletions;
  - 1% branch creations; and
  - 0% argument creations.
- **The percentage of operations on each generation before generation 6 was**
  - 78.0% one-offspring crossovers;
  - 10% reproductions;
  - 1% mutations;
  - 5.0% branch duplications;
  - 1% branch deletions;
  - 5.0% branch creations; and
  - 0% argument creations.

CONTROL PARAMETERS (FOR 96 DB AMPLIFIER) – CONTINUED

- **Maximum of H_{rpb} = 300 points (functions and terminals) for each result-producing branch**

- **Maximum of H_{adf} = 200 points (functions and terminals) for each function-defining branch**

- **The maximum number of automatically defined functions is 4.**

- **The number of arguments for each automatically defined function is 1.**

- **Secondary parameters are default values in Koza 1994 (appendix D)**

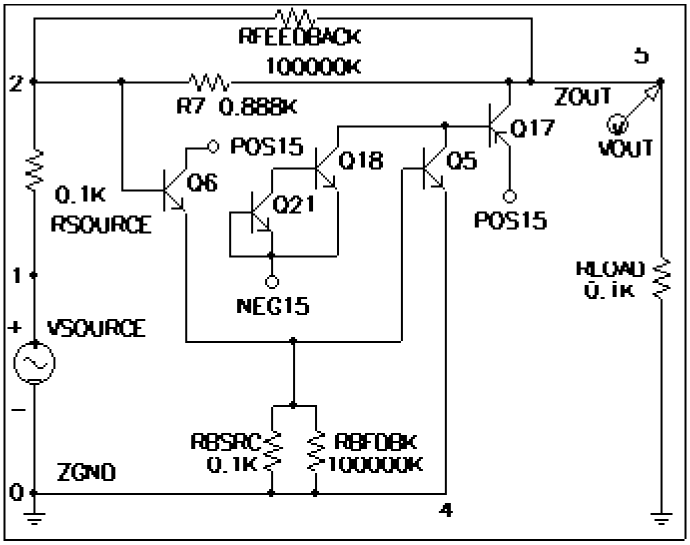
TERMINATION CRITERION AND RESULTS DESIGNATION (FOR 96 DB AMPLIFIER)

- **Manual intervention in lieu of pre-established termination criterion**

- **Best-so-far individual is designated as the result of the run**

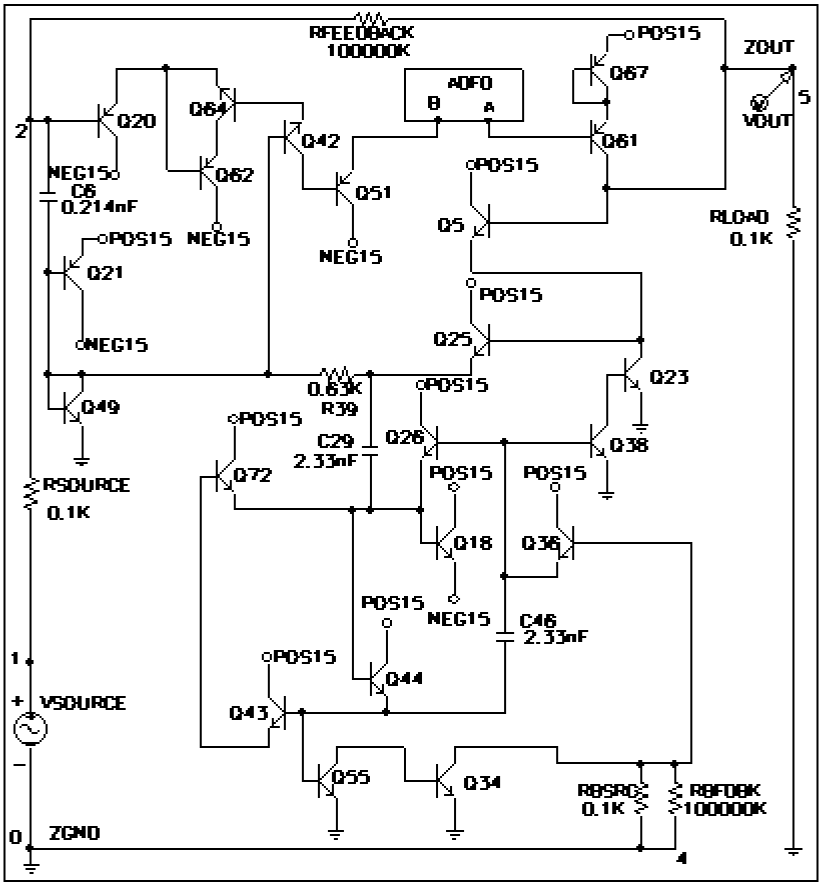
BEST CIRCUIT OF GENERATION 0

(96 DB)

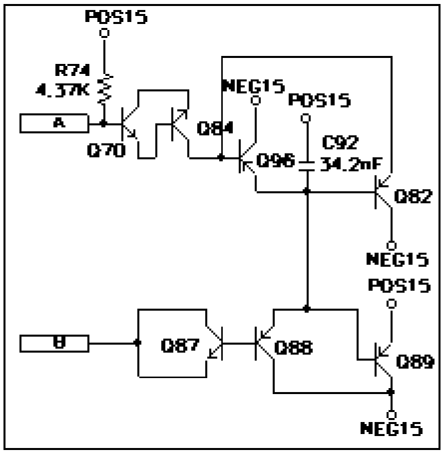


BEST CIRCUIT OF GENERATION 42

(96 DB)

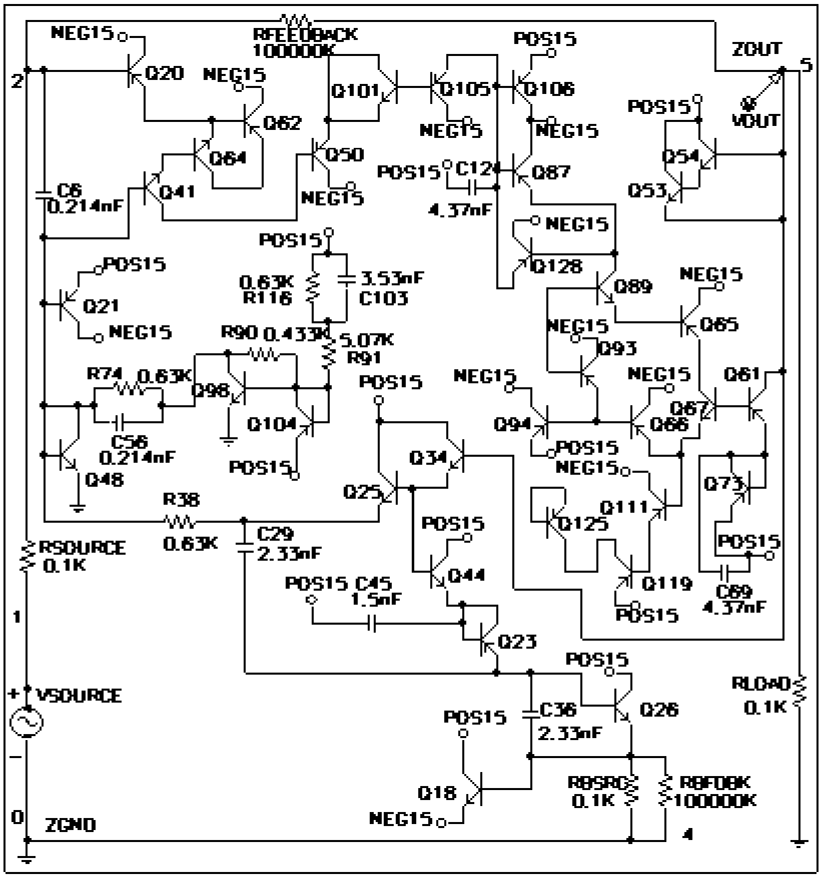


ADF0



BEST CIRCUIT OF GENERATION 50

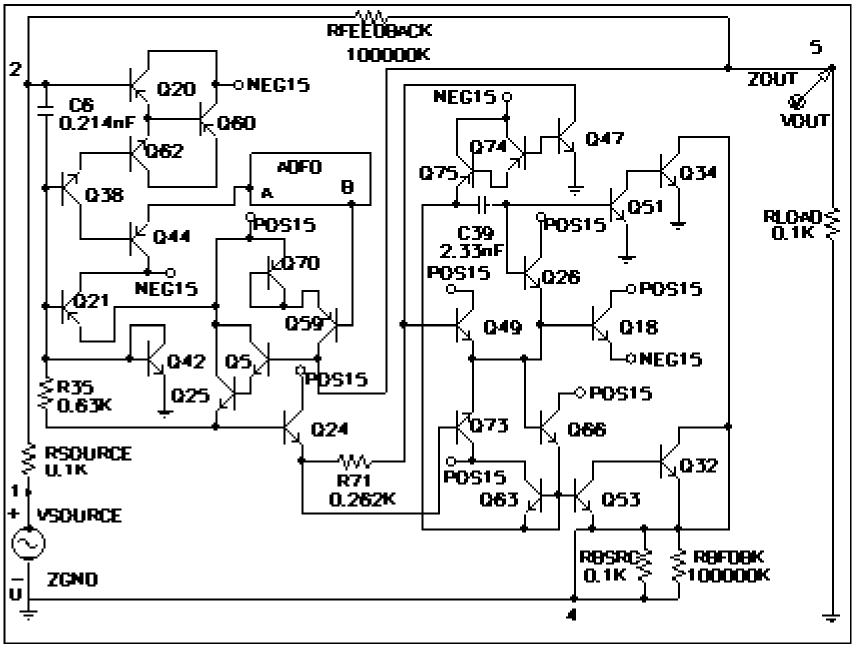
(96 DB)



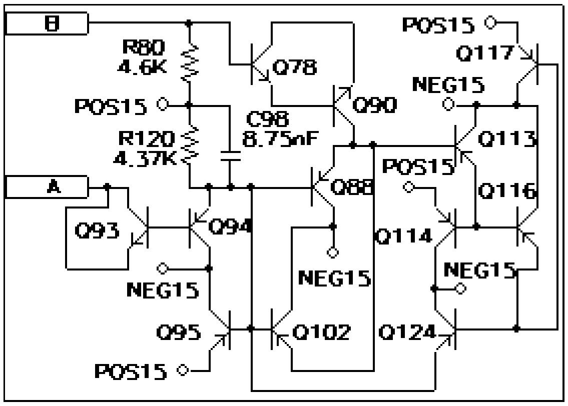
NO ADFS

BEST CIRCUIT OF GENERATION 86

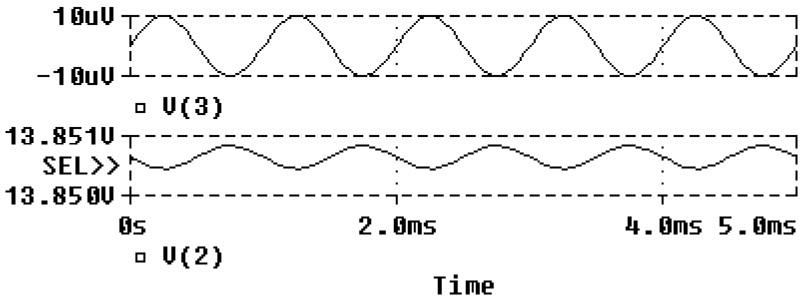
(96 DB)

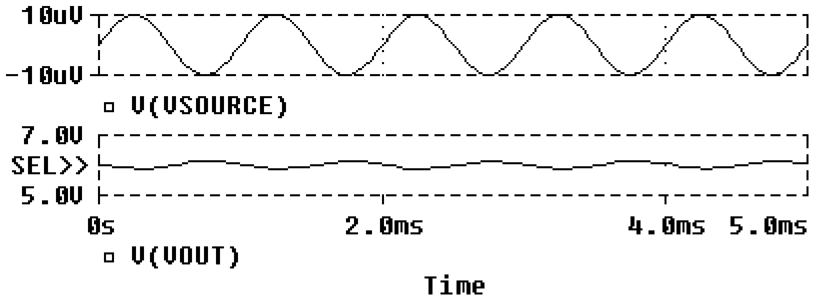


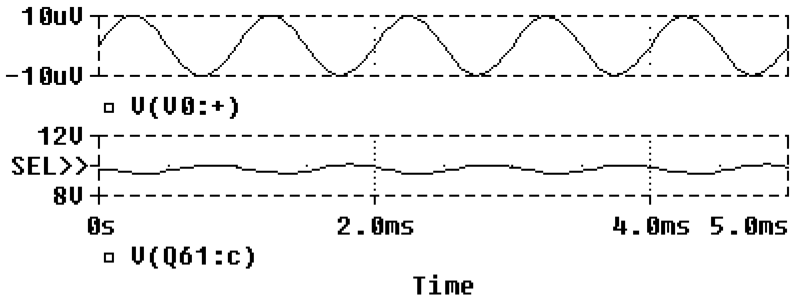
ADF0

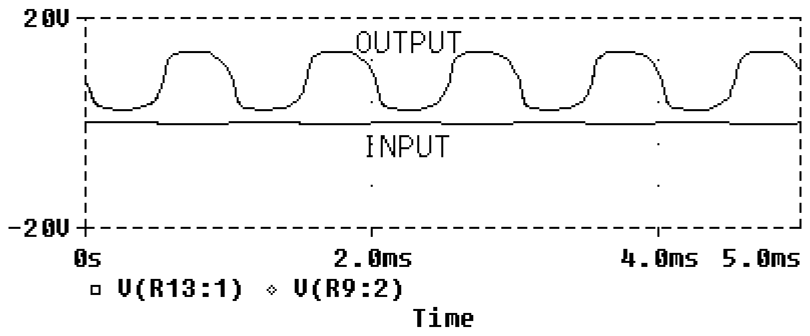


TIME DOMAIN – BEST OF GENS 0, 42, 50, 86 (96 DB)

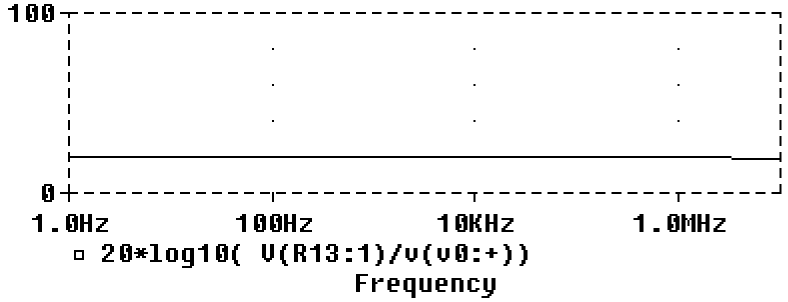


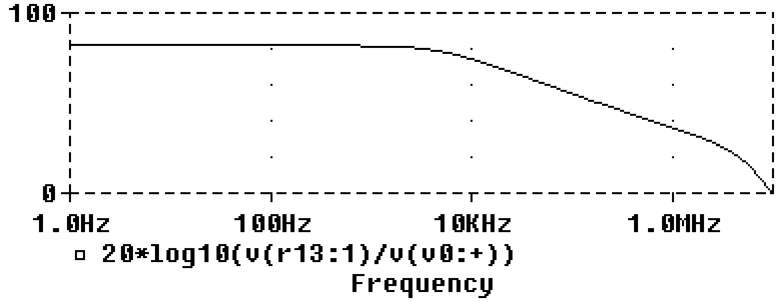


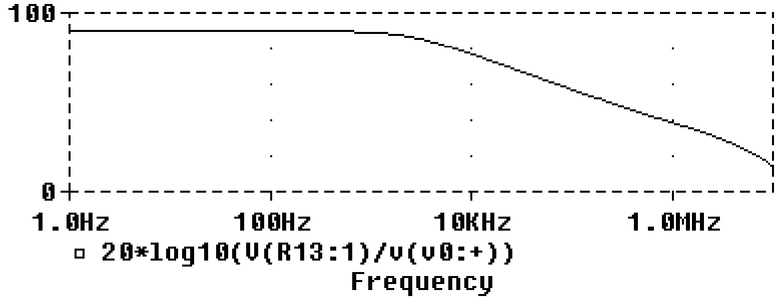


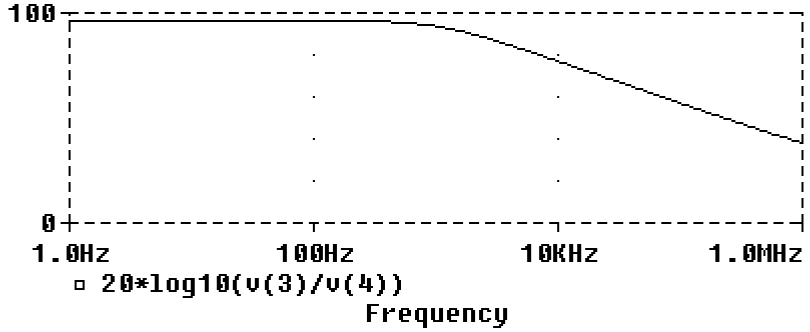


AC SWEEPS – BEST OF GENS 0, 42, 50, 86 (96 DB)

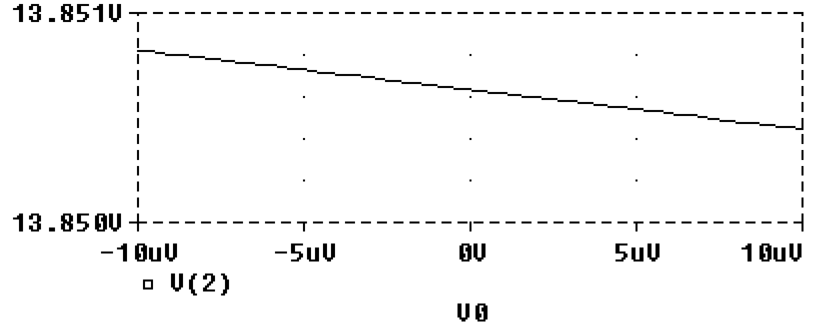


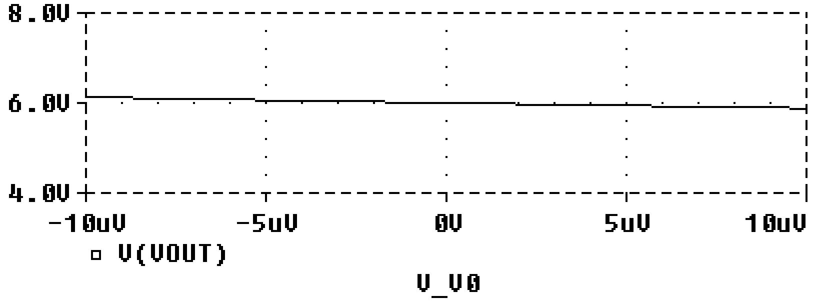


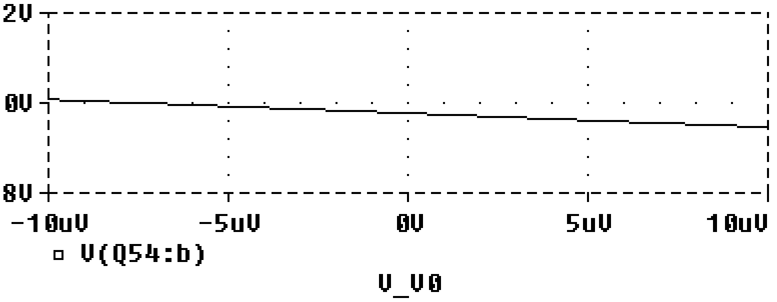


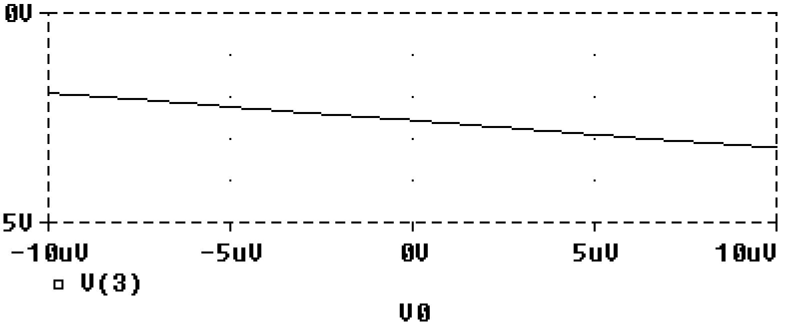


DC SWEEP – BEST OF GENS 0, 42, 50, 86 (96 DB)

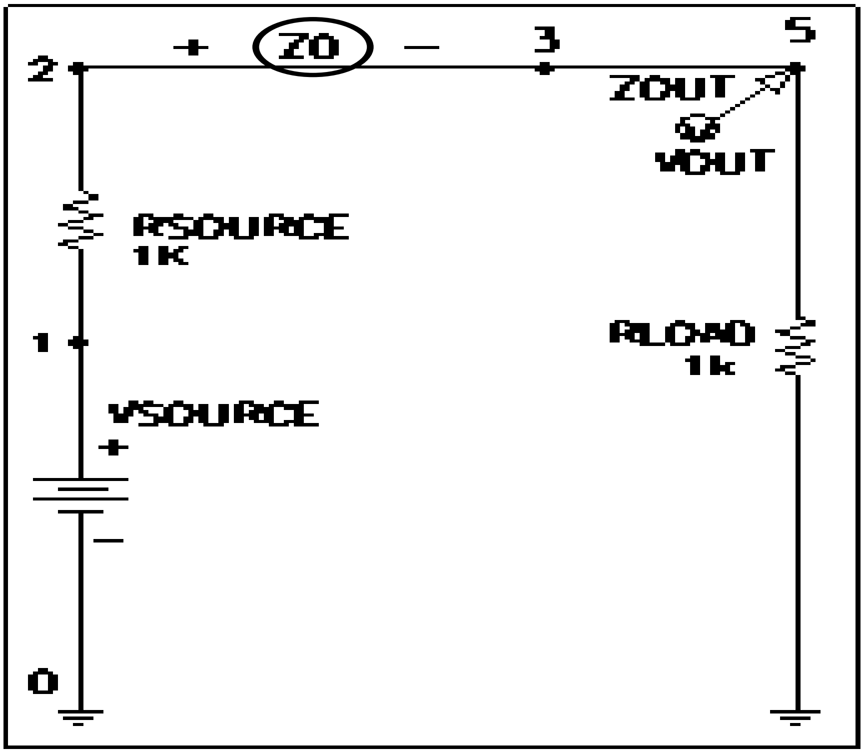








ONE-INPUT, ONE-OUTPUT EMBRYO WITH ONE WRITING HEAD (ONE MODIFIABLE WIRE) FOR COMPUTATIONAL CIRCUITS



ARCHITECTURE OF CIRCUIT-CONSTRUCTING PROGRAM TREE FOR COMPUTATIONAL CIRCUITS

- **The circuit-constructing program tree has one result-producing branch (`RPB0`).**

- **The circuit-constructing program tree has no automatically defined functions.**

FUNCTION AND TERMINAL SETS OF RESULT PRODUCING BRANCHES FOR COMPUTATIONAL CIRCUITS

- **For the result-producing branch, the function set, $F_{ccs\text{-}rpb}$, for each construction-continuing subtree is**

$F_{ccs\text{-}rpb}$ = {`R`, `SERIES`, `PARALLEL0`, `PARALLEL1`, `FLIP`, `NOP`, `NEW_T_GND_0`, `NEW_T_GND_1`, `NEW_T_POS_0`, `NEW_T_POS_1`, `NEW_T_NEG_0`, `NEW_T_NEG_1`, `PAIR_CONNECT_0`, `PAIR_CONNECT_1`, `Q_D_NPN,` `Q_D_PNP`, `Q_3_NPN0`, ..., `Q_3_NPN11`, `Q_3_PNP0`, ..., `Q_3_PNP11`, `Q_POS_COLL_NPN`, `Q_GND_EMIT_NPN`, `Q_NEG_EMIT_NPN`, `Q_GND_EMIT_PNP`, `Q_POS_EMIT_PNP`, `Q_NEG_COLL_PNP`}

- **For the result-producing branch, the function set, $F_{ccs\text{-}rpb}$, for each construction-continuing subtree is**

$T_{ccs\text{-}rpb}$ = {`END`, `SAFE_CUT`}.

FUNCTION AND TERMINAL SETS FOR ARITHMETIC-PERFORMING SUBTREES (FOUND IN BOTH RBPs AND ADFs) FOR COMPUTATIONAL CIRCUITS

- **The terminal set, T_{aps}, for each arithmetic-performing subtree consists of**

$T_{aps} = \{\Re\}$,

where $\Re$ represents floating-point random constants from –1.0 to +1.0.

- **The function set, F_{aps}, for each arithmetic-performing subtree is,**

$F_{aps} = \{+, -\}$.

FITNESS MEASURE (FOR COMPUTATIONAL CIRCUITS)

- **The target voltage is the square root, cube root, square, cube of the input voltage – depending on the particular computational circuit desired.**

- **The SPICE simulator is requested to perform a DC sweep analysis at 21 equidistant voltages between –250 mV and +250 mV for the cube root, square, and cube functions**
  - but only 0 mV to +500 mV for the square root
- **Fitness is the sum, over these 21 fitness cases, of the absolute weighted deviation between the actual value of the voltage that is produced by the circuit at the probe point VOUT and the target value for voltage. The smaller the value of fitness, the better.**

FITNESS MEASURE (FOR COMPUTATIONAL CIRCUITS)

*** The fitness measure does not penalize output voltages that perfectly match the target voltages; it slightly penalizes every acceptable deviation from the target voltage; and it heavily penalizes every unacceptable deviation.**

- If the output voltage is within 1% of the target voltage value for a particular fitness case, the absolute value of the deviation is weighted by 1 for that fitness case.
- If the output voltage is not within 1% of the target voltage value, the deviation is weighted by 10 for that fitness case.

- **This arrangement reflects the fact that a deviation of 1% from the ideal voltage is acceptable, but greater deviations are not.**

- **Unsimulatable programs = 10^8 penalty**

CONTROL PARAMETERS FOR COMPUTATIONAL CIRCUITS

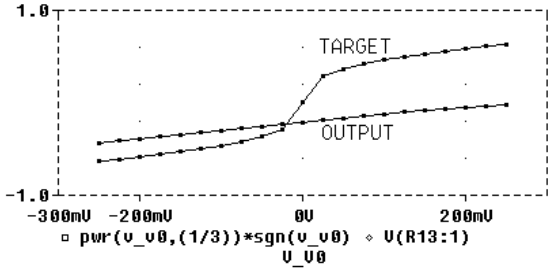
- **Population size, *M*, of 640,000**
- **Maximum number of generations, *G*, is set to be meaninglessly large**
- **Maximum of H_{rpb} = 600 points (functions and terminals) for each result-producing branch**
- **For each generation**
  - 10% reproductions
  - 1% mutations
  - 89% crossovers
  - No architecture-altering operations
- **Secondary parameters are default values in Koza 1994 (appendix D)**

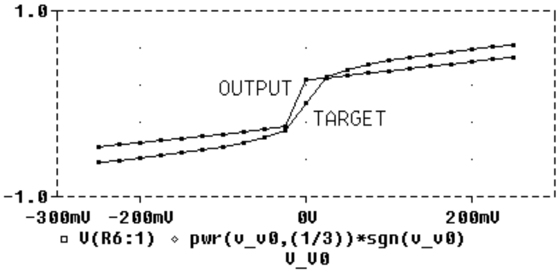
TERMINATION CRITERION AND RESULTS DESIGNATION FOR COMPUTATIONAL CIRCUITS

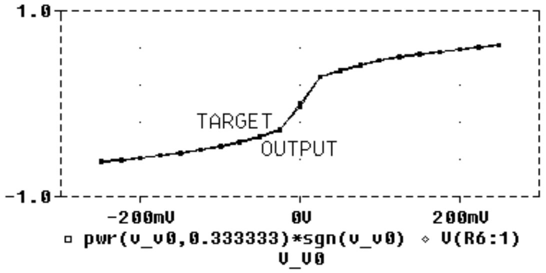
- **Manual intervention in lieu of pre-established termination criterion**

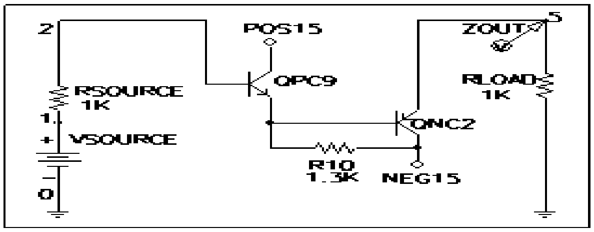
- **Best-so-far individual is designated as the result of the run**

RESULTS FOR CUBE ROOT CIRCUIT FROM GENERATIONS 0, 17, 60

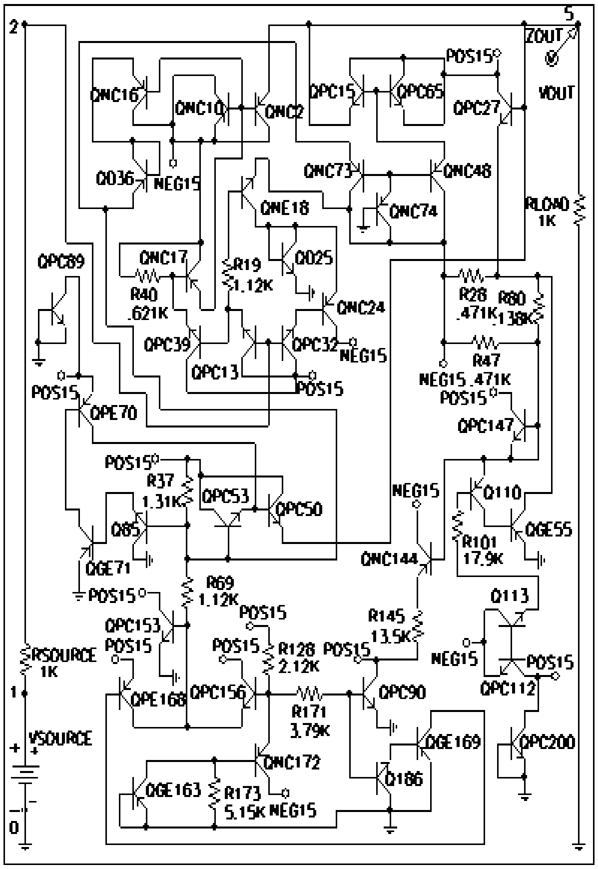




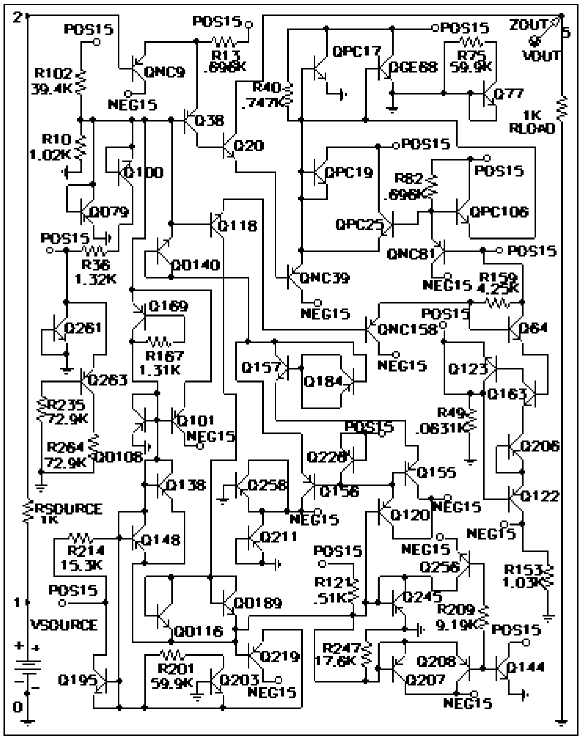




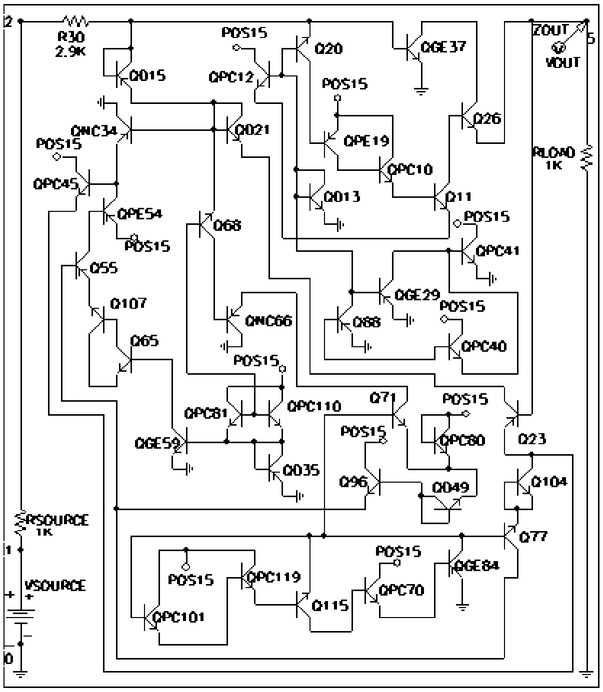
CUBE ROOT CIRCUIT FROM GENERATIONS 60



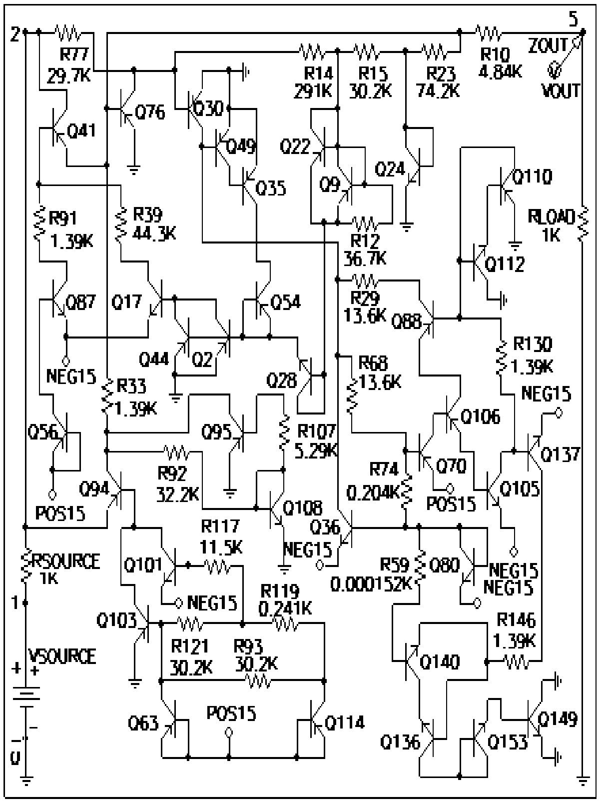
EVOLVED SQUARE ROOT CIRCUIT



EVOLVED SQUARING CIRCUIT



EVOLVED CUBING CIRCUIT



FUNCTION AND TERMINAL SETS

- **For RPBs, function set for construction-continuing subtree**

$F_{ccs\text{-}rpb}$ = {**`ADF0`**, **`ADF1`**, **`ADF2`**, **`ADF3`**, `C`, `L`, `SERIES`, `PARALLEL0`, `FLIP`, `NOP`, `THGND`, `CUT`, `THVIA0`, `THVIA1`, `THVIA2`, `THVIA3`, `THVIA4`, `THVIA5`, `THVIA6`, `THVIA7`}

- **For RPBs, terminal set for construction-continuing subtree**

$T_{ccs\text{-}rpb}$ = {`END`, `CUT`}

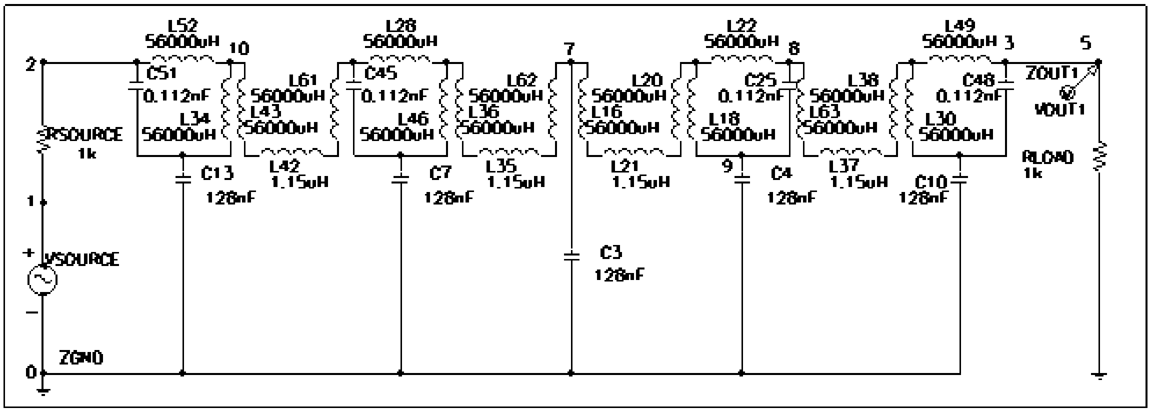
- **For ADFs, function set for construction-continuing subtree**

F_{ccs} = {`C`, `L`, `SERIES`, `PARALLEL0`, `FLIP`, `NOP`, `THGND`, `CUT`, `THVIA0`, `THVIA1`, `THVIA2`, `THVIA3`, `THVIA4`, `THVIA5`, `THVIA6`, `THVIA7`}

- **For ADFs, the terminal set for construction-continuing subtree**

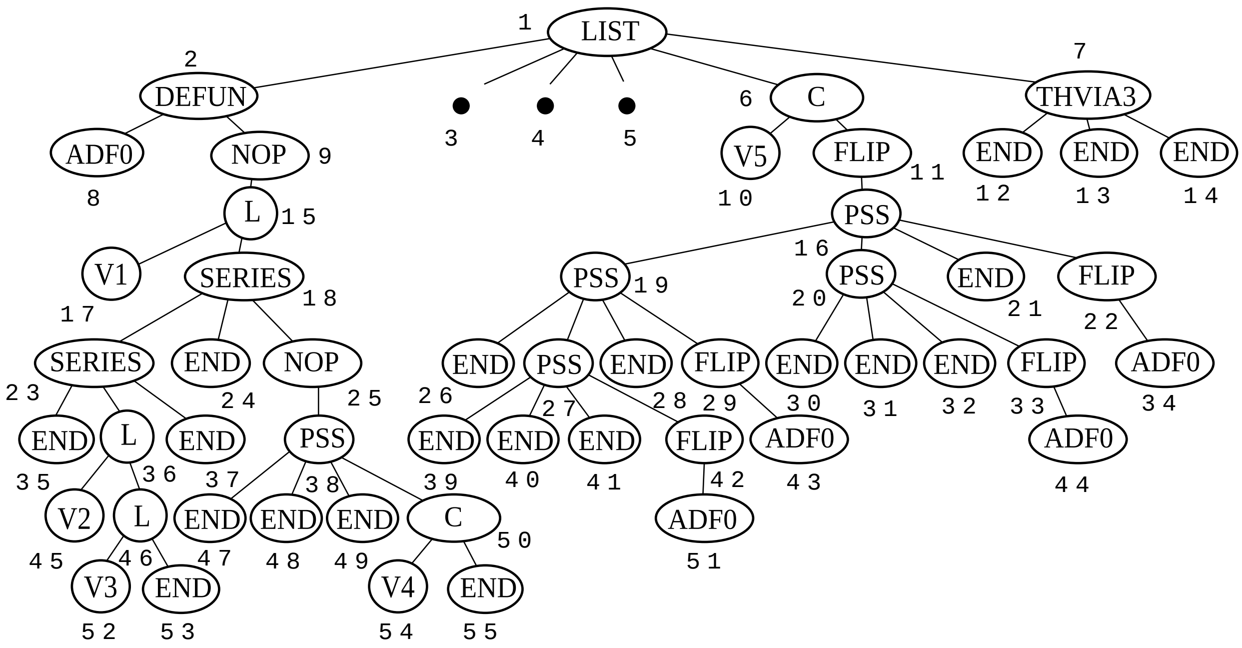
$T_{ccs\text{--}adf}$ = {`END`, `CUT`}

BEST-OF-RUN CIRCUIT FROM GENERATION 35 USING ADFs

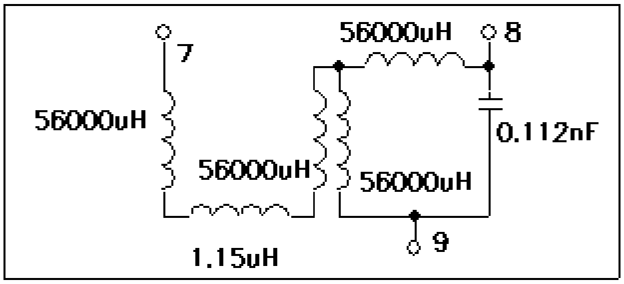


NOTE SYMMETRY

BEST-OF-RUN PROGRAM TREE FROM GENERATION 35 USING ADFs



EDITED VERSION OF `ADF0` FROM BEST-OF-RUN PROGRAM TREE FROM GENERATION 35 USING ADFs



FUNCTION AND TERMINAL SETS USING ARCHITECTURE-ALTERING OPERATIONS

- **For RPBs, function set for construction-continuing subtree**

$F_{ccs\text{-}rpb}$ = {`C`, `L`, `SERIES`, `PARALLEL0`, `FLIP`, `NOP`, `THGND`, `CUT`, `THVIA0`, `THVIA1`, `THVIA2`, `THVIA3`, `THVIA4`, `THVIA5`, `THVIA6`, `THVIA7`}

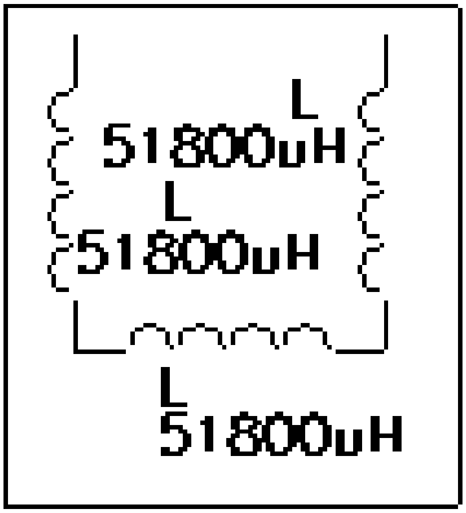
- **For RPBs, terminal set for construction-continuing subtree**

$T_{ccs\text{-}rpb}$ = {`END`, `CUT`}

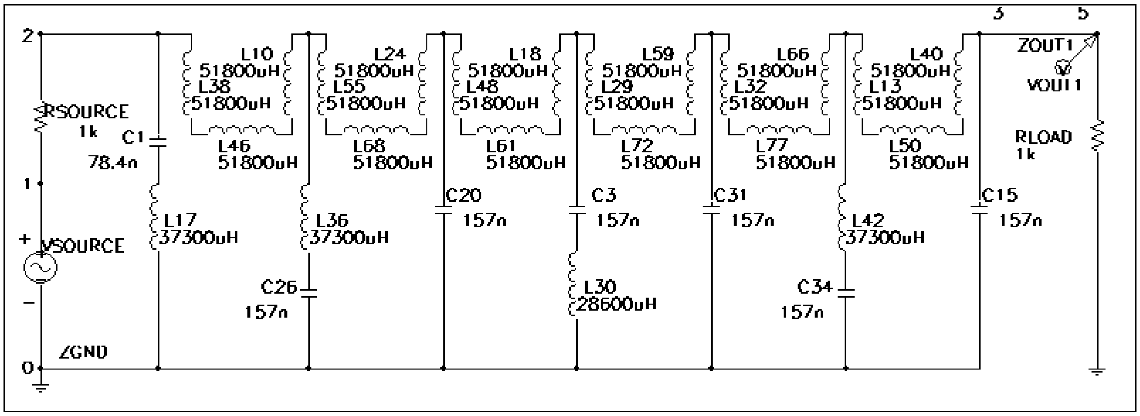
- **For the new ADFs, the set of potential new functions, $F_{ccs\text{-}pot}$, is**

$F_{ccs\text{-}pot}$ = {`ADF0`, `ADF1`, `ADF2`, `ADF3`}

AUTOMATICALLY DEFINED FUNCTION `ADF0` OF BEST-OF-RUN CIRCUIT FROM GENERATION 77 USING ARCHITECTURE-ALTERING OPERATIONS

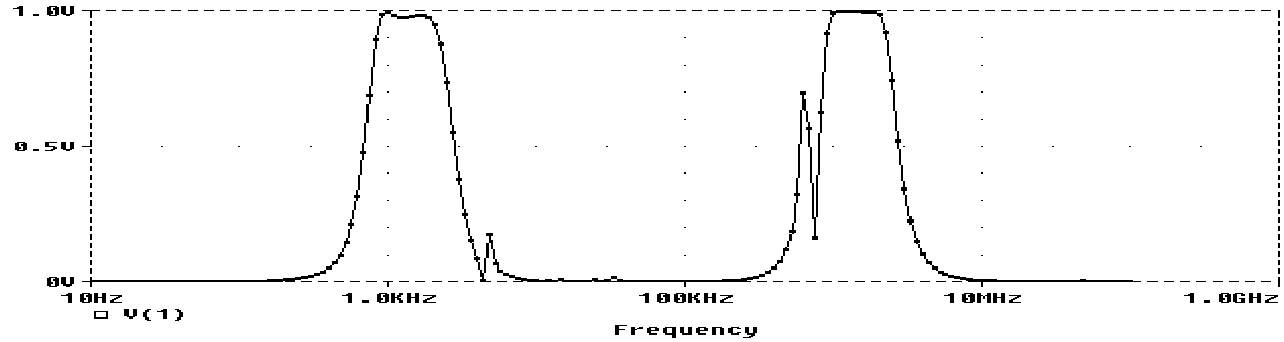


BEST-OF-RUN CIRCUIT FROM GENERATION 77 USING ARCHITECTURE-ALTERING OPERATIONS

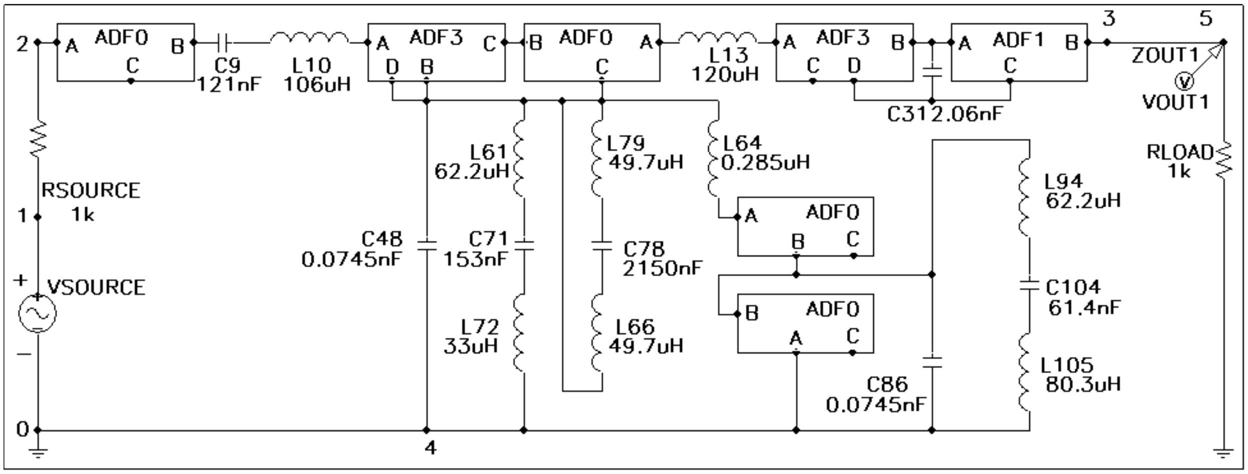


EVOLVING A DOUBLE-BANDPASS FILTER USING ARCHITECTURE-ALTERING OPERATIONS

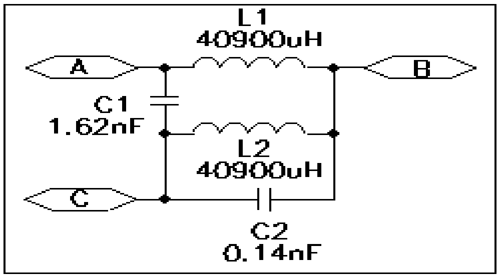
GENERATION 89 – FREQUENCY DOMAIN BEHAVIOR OF THE BEST-OF-RUN CIRCUIT



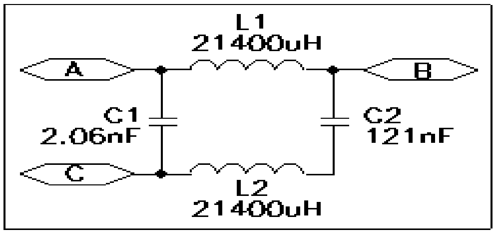
GENERATION 89 – BEST-OF-RUN CIRCUIT



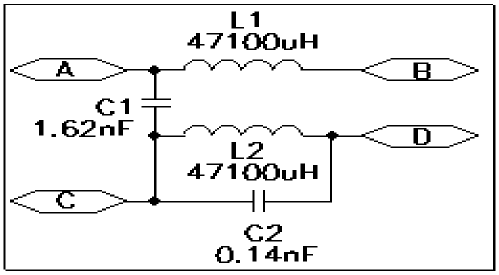
THREE-PORTED QUADRUPLY-CALLED **ADF0**



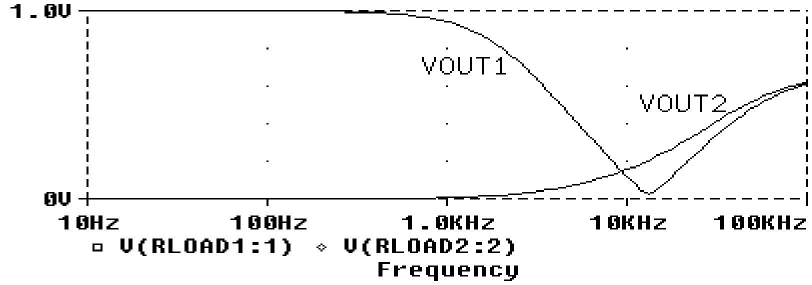
THREE-PORTED **ADF1**

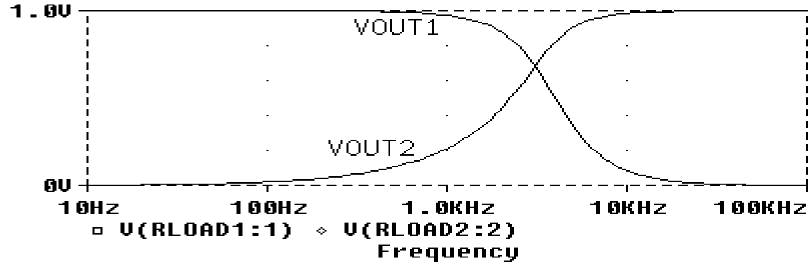


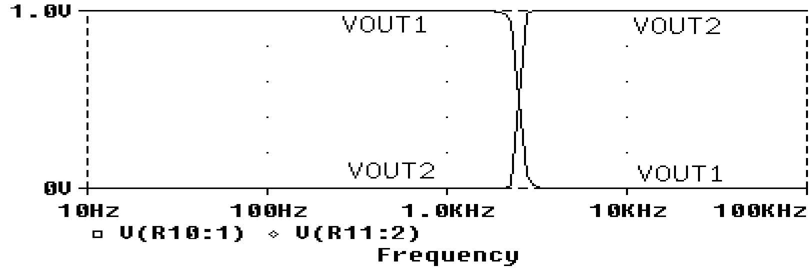
FOUR-PORTED TWICE-CALLED **ADF3**



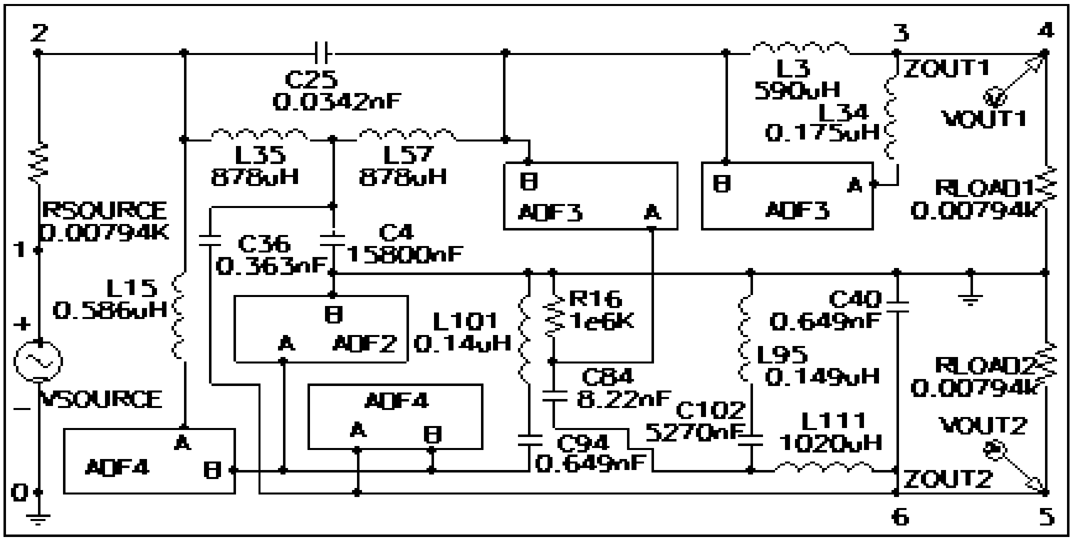
FREQUENCY DOMAIN BEHAVIOR OF BEST CIRCUIT OF GENERATIONS 0, 8, AND 158 FOR A TWO-BAND CROSSOVER FILTER



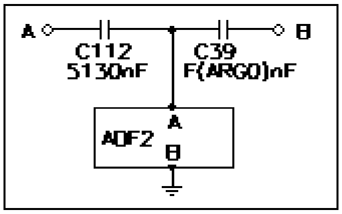




BEST CIRCUIT OF GENERATION 158



- **ADF3 supplies one PARAMETERIZED capacitor C39 whose value is determined by ADF3's dummy variable, ARG0.**

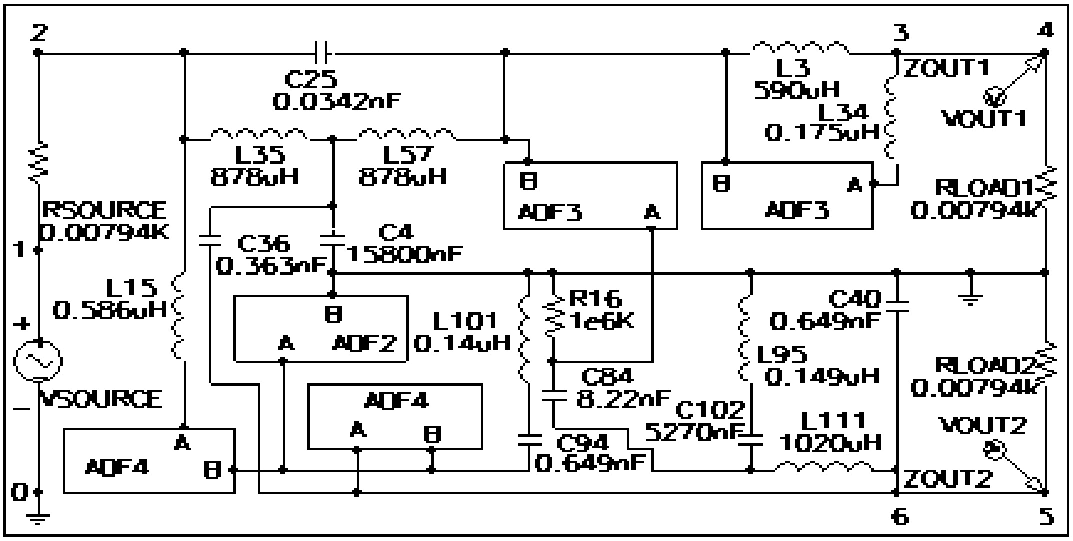


- **ADF3 supplies one unparameterized 5,130 uF capacitor C112.**
- **ADF3 has one hierarchical reference to ADF2 (which, in turn, supplies one unparameterized 259 μH inductor).**

BEST CIRCUIT OF GENERATION 158 — CONTINUED

- **The combined effect of `ADF3` is to supply two capacitors (one of which is parameterized) and one inductor.**
- **`ADF3` has three ports and is called is called once by `RPB0` and `RPB1`.**

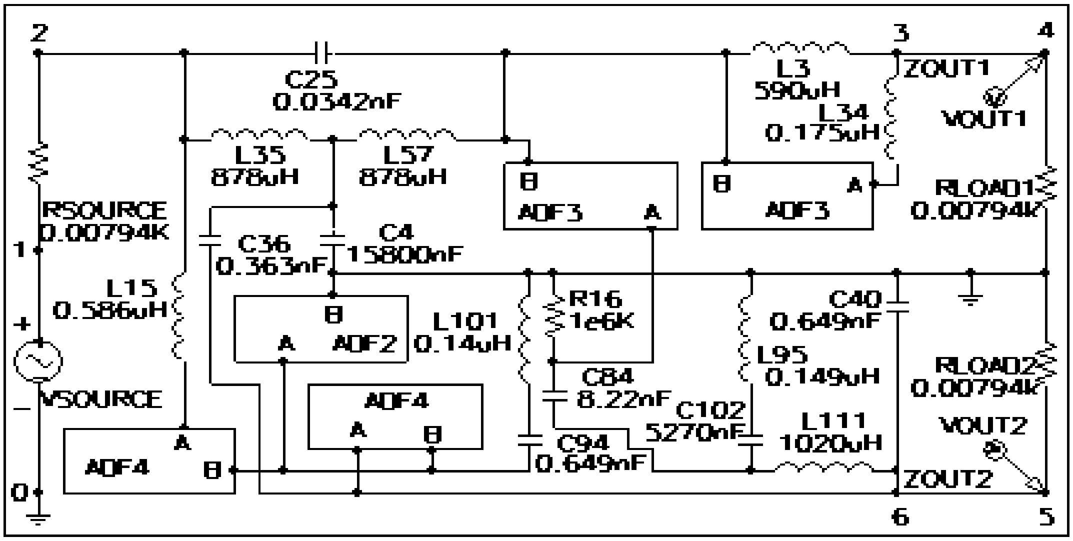
BEST CIRCUIT OF GENERATION 158 – CONTINUED



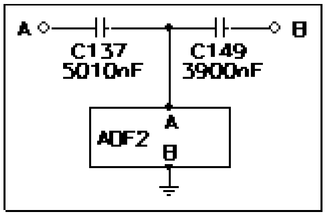
- **`ADF0` and `ADF1` are not called at all.**

- **`ADF2` has two ports and supplies one unparameterized 259 μH inductor `L147`. `ADF2` is called a total of five times – one time by `RPB2` directly, twice hierarchically by `ADF3` (which is called once by `RPB0` and `RPB1`), and twice hierarchically by `ADF4` (called by `RFP2`). Its `ARG0` plays no role.**

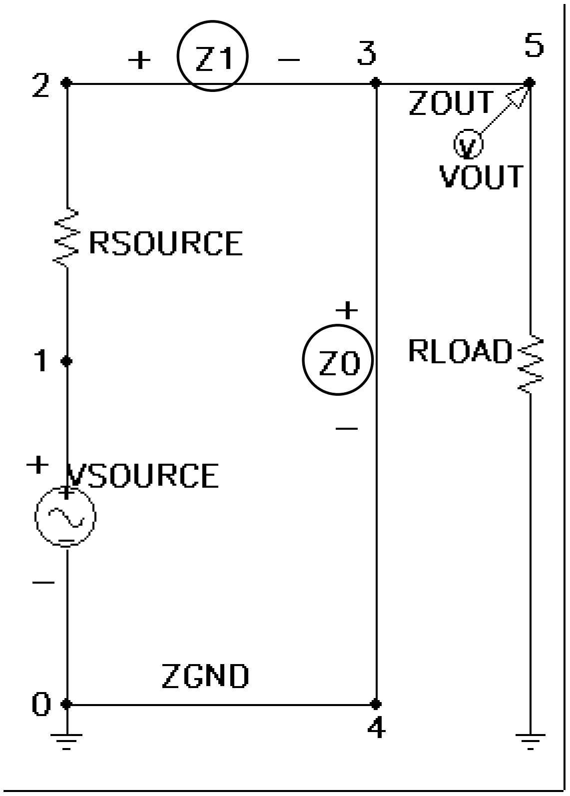
BEST CIRCUIT OF GENERATION 158 – CONTINUED



- **ADF4 has three ports and supplies one unparameterized 3,900 uF capacitor C137 and one unparameterized 5,010 uF capacitor C149. ADF4 has one hierarchical reference to ADF2 (which, in turn, supplies one unparameterized 259 μH inductor). Thus, the combined effect of ADF4 is to supply two capacitors and one inductor.**



ONE-INPUT, ONE-OUTPUT EMBRYONIC ELECTRICAL CIRCUIT FOR LOWPASS FILTER



ARCHITECTURE OF CIRCUIT-CONSTRUCTING PROGRAM TREE FOR LOWPASS FILTER

- **Two result-producing branches (`RPB0` and `RPB1`) joined by a connective `LIST` function**

- **Four automatically defined functions (`ADF0`, `ADF1`, `ADF2`, and `ADF3`)**

- **The circuit-constructing program tree has six branches joined by a connective `LIST` function**

FUNCTION AND TERMINAL SETS FOR LOWPASS FILTER

- **For RPBs, function set for construction-continuing subtree**

$F_{ccs\text{-}rpb}$ = {**`ADF0`**, **`ADF1`**, **`ADF2`**, **`ADF3`**, `C`, `L`, `SERIES`, `PARALLEL0`, `FLIP`, `NOP`, `THGND`, `CUT`, `THVIA0`, `THVIA1`, `THVIA2`, `THVIA3`, `THVIA4`, `THVIA5`, `THVIA6`, `THVIA7`}

- **For RPBs, terminal set for construction-continuing subtree**

$T_{ccs\text{-}rpb}$ = {`END`, `CUT`}

- **For ADFs, function set for construction-continuing subtree**

F_{ccs} = {`C`, `L`, `SERIES`, `PARALLEL0`, `FLIP`, `NOP`, `THGND`, `CUT`, `THVIA0`, `THVIA1`, `THVIA2`, `THVIA3`, `THVIA4`, `THVIA5`, `THVIA6`, `THVIA7`}

- **For ADFs, the terminal set for construction-continuing subtree**

$T_{ccs\text{--}adf}$ = {`END`, `CUT`}

FITNESS MEASURE FOR LOWPASS FILTER

- **Our modified version of SPICE (217,000 lines of C source code) gives output values at probe point VOUT**
- **101 frequency values chosen over five decades (from 1 to 100,000 Hz) with each decade divided into 20 parts (using a logarithmic scale).**
  - do not penalize ideal values
  - slightly penalize acceptable deviations
  - heavily penalize unacceptable deviations
- **Fitness is**

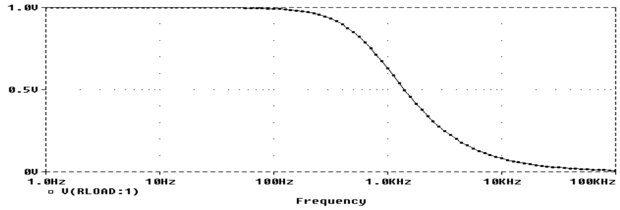
$$\boldsymbol{F(t)} = \sum_{i=0}^{100} [W(f_i)d(f_i)]$$

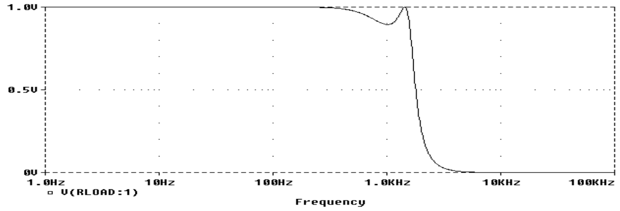
  - $f(i)$ is the frequency (in Hertz) of fitness case i
  - $d(x)$ is the difference between the target and observed values at frequency (in Hertz) of fitness case i
  - $W(y,x)$ is the weighting at frequency x

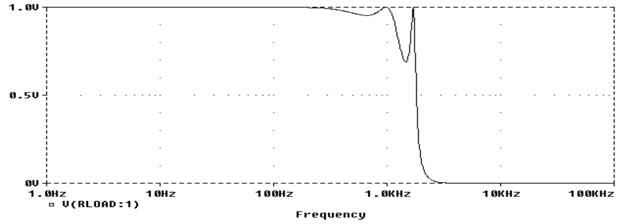
FITNESS – CONTINUED

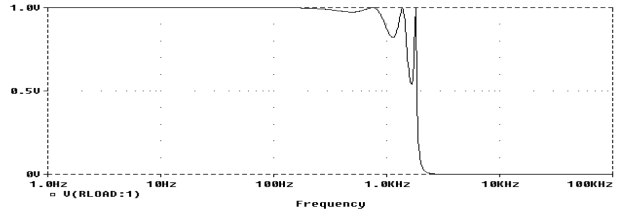
- **61 points in the 3-decade interval from 1 Hz to 1,000 Hz**
  - For voltage equaling the ideal value of 1.0 volts, the deviation is **0.0**
  - For voltage between 970 and 1,000 millivolts, the absolute value of the deviation from 1,000 millivolts is weighted by **1.0**
  - For voltage less than 970 millivolts, the absolute value of the deviation from 1,000 millivolts is weighted by a factor of **10.0**
- **35 points from 2,000 Hz to 100,000 Hz**
  - For voltage equaling the ideal value of 0.0 volts, the deviation is **0.0**
  - For voltage between 0 millivolts and 1 millivolt, the absolute value of the deviation from 0 millivolts is weighted by **1.0**
  - For voltage above 1 millvolt, the absolute value of the deviation from 0 millivolts is weighted by factor of **10.0**
- **5 "don't care" points between 1,000 Hz and 2,000 Hz**

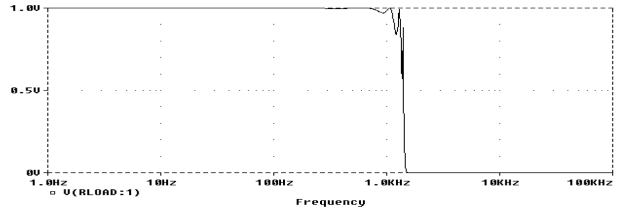
GENERATIONS 0, 9, 16, 20, 31, AND 35

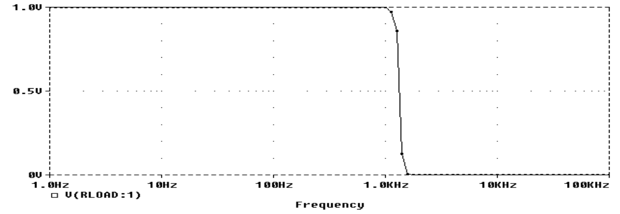






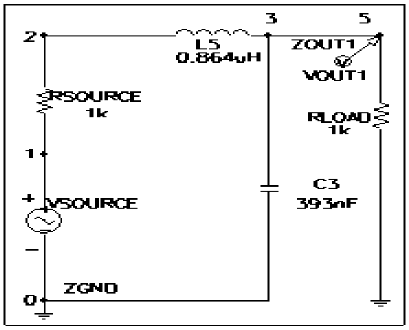




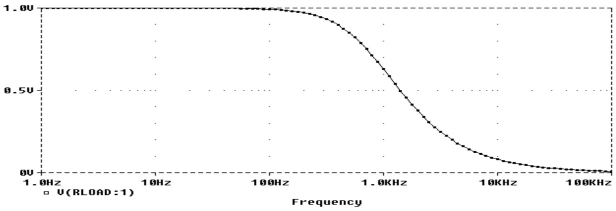


LOWPASS FILTER USING ADFs - RUN B

GENERATION 0 – ONE-RUNG LADDER

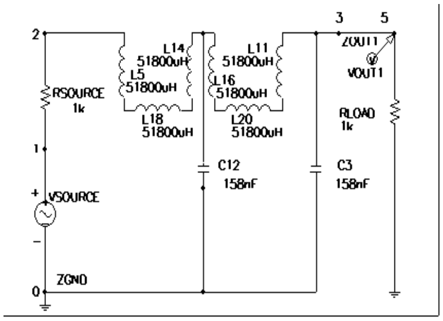


BEHAVIOR IN FREQUENCY DOMAIN

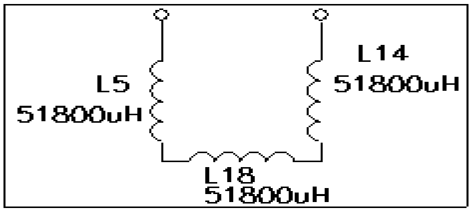


LOWPASS FILTER USING ADFs - RUN B

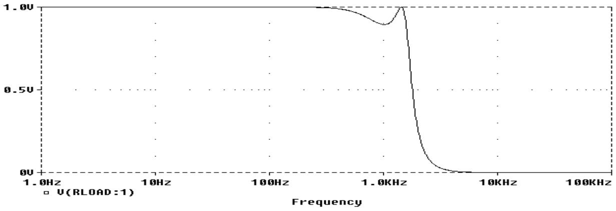
GENERATION 9 - TWO-RUNG LADDER



TWICE-CALLED TWO-PORTED `ADF0`

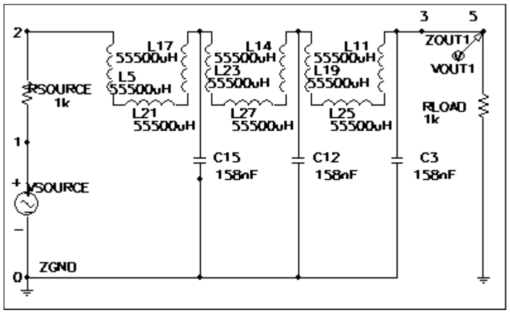


BEHAVIOR IN FREQUENCY DOMAIN

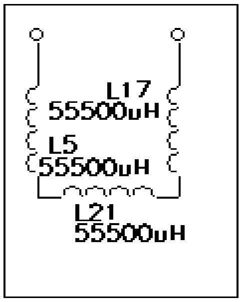


LOWPASS FILTER USING ADFs - RUN B

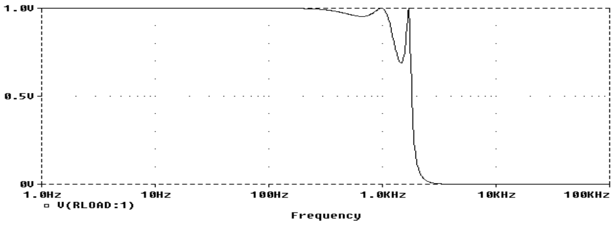
GEN 16 – THREE-RUNG LADDER



THRICE-CALLED TWO-PORTED `ADF0`

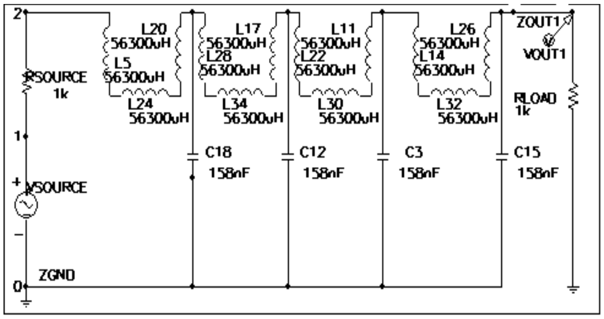


BEHAVIOR IN FREQUENCY DOMAIN

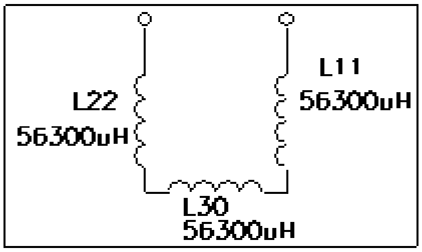


LOWPASS FILTER USING ADFs - RUN B

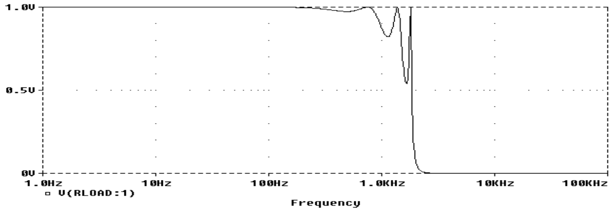
GEN 20 – FOUR-RUNG LADDER



QUADRUPLY-CALLED TWO-PORTED ADF0

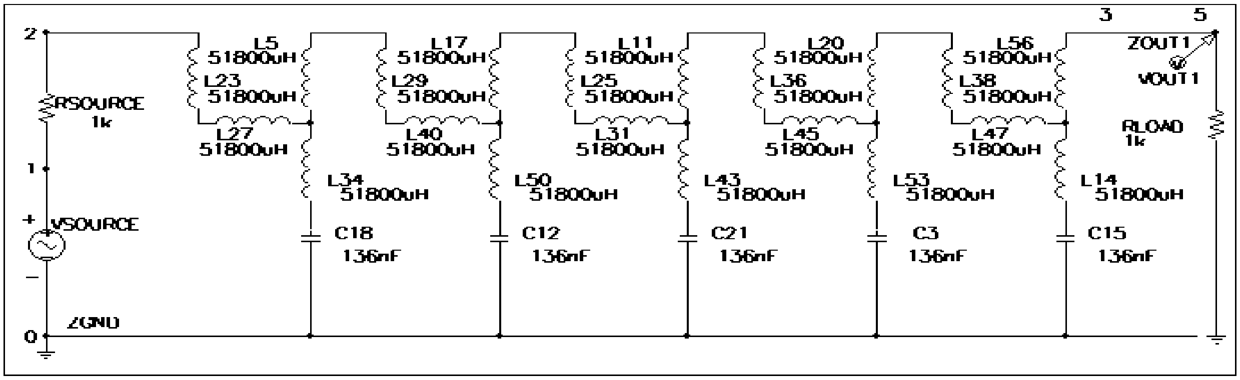


BEHAVIOR IN FREQUENCY DOMAIN

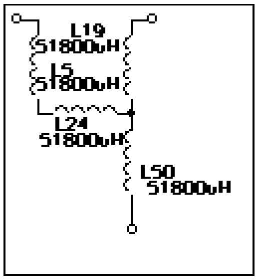


LOWPASS FILTER USING ADFs - RUN B

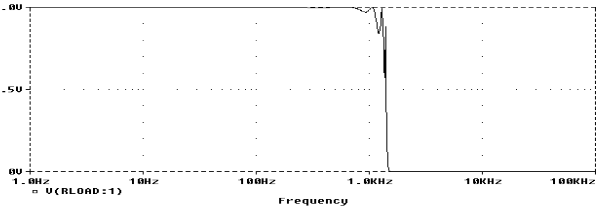
GENERATION 31 — TOPOLOGY OF CAUER (ELLIPTIC) FILTER



QUINTUPLY-CALLED THREE-PORTED `ADF0`



BEHAVIOR IN FREQUENCY DOMAIN

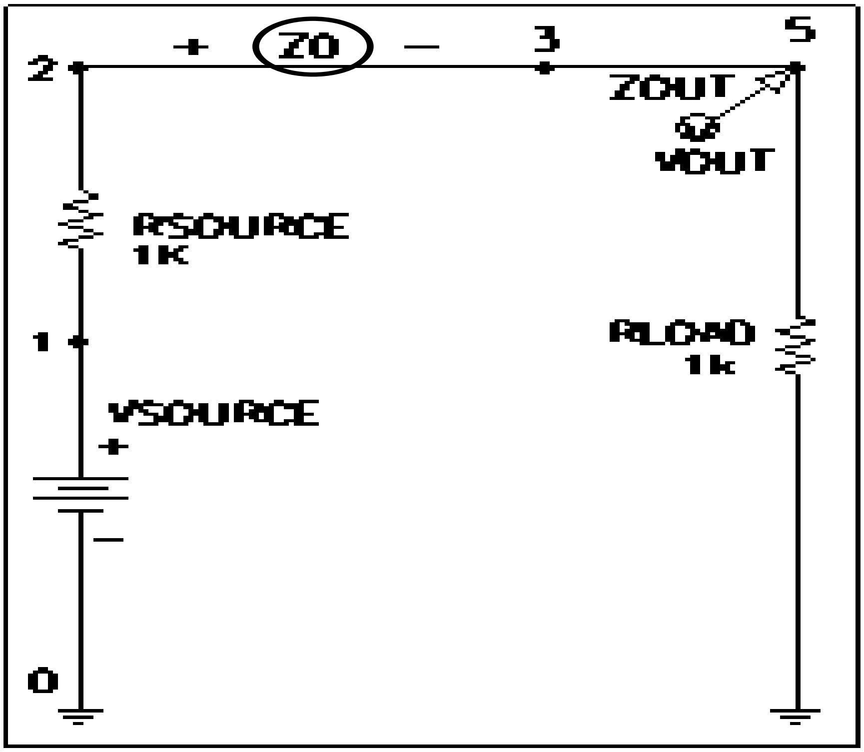


CAUER (ELLIPTIC) FILTERS

> **"Cauer first used his new theory in solving a filter problem for the German telephone industry. His new design achieved specifications with one less inductor than had ever been done before. The world first learned of the Cauer method not through scholarly publication but through a patent disclosure, which eventually reached the Bell Laboratories. Legend has it that the entire Mathematics Department of Bell Laboratories spent the next two weeks at the New York Public library studying elliptic functions. Cauer had studied mathematics under Hilbert at Goettingen, and so elliptic functions and their applications were familiar to him."**

– from Van Valkenburg *Analog Filter Design* (1982, page 379)

ONE-INPUT, ONE-OUTPUT EMBRYO WITH ONE WRITING HEAD (ONE MODIFIABLE WIRE) FOR THREE-WAY ANALOG SOURCE IDENTIFICATION PROBLEM



ARCHITECTURE OF CIRCUIT-CONSTRUCTING PROGRAM TREE FOR THREE-WAY ANALOG SOURCE IDENTIFICATION PROBLEM

- **The circuit-constructing program tree has one result-producing branch (`RPB0`).**

- **The circuit-constructing program tree has no automatically defined functions.**

FUNCTION AND TERMINAL SETS OF RESULT PRODUCING BRANCH FOR THREE-WAY ANALOG SOURCE IDENTIFICATION PROBLEM

- **For the result-producing branch, the function set, $F_{ccs\text{-}rpb}$, for each construction-continuing subtree is**

$F_{ccs\text{-}rpb}$ = {`R`, `L`, `C`, `SERIES`, `PARALLEL0`, `PARALLEL1`, `FLIP`, `NOP`, `T_PAIR_CONNECT_0`, `T_PAIR_CONNECT_1`}

- **For the result-producing branch, the function set, $F_{ccs\text{-}rpb}$, for each construction-continuing subtree is**

$T_{ccs\text{-}rpb}$ = {`END`}.

FUNCTION AND TERMINAL SETS FOR ARITHMETIC-PERFORMING SUBTREES (USED IN BOTH RBPs AND ADFs) FOR THREE-WAY ANALOG SOURCE IDENTIFICATION PROBLEM

- **The function set, F_{aps}, for each arithmetic-performing subtree is,**

$F_{aps} = \{+, -\}$.

- **The terminal set, T_{aps}, for each arithmetic-performing subtree consists of**

$T_{aps} = \{\Re\}$,

where $\Re$ represents floating-point random constants from –1.0 to +1.0.

FITNESS MEASURE FOR THREE-WAY ANALOG SOURCE IDENTIFICATION PROBLEM

- **Voltage VOUT is probed at node 5 and the circuit is simulated in the frequency domain.**

- **SPICE is requested to perform an AC small signal analysis and to report the circuit's behavior for each of 101 frequency values chosen over four decades of frequency (between 1 and 10,000 Hz). Each decade is divided into 25 parts (using a logarithmic scale).**

- **Fitness is measured in terms of the sum, over these 101 fitness cases, of the absolute weighted deviation between the actual value of the output voltage at the probe point VOUT and the target value for voltage.**

FITNESS MEASURE FOR THREE-WAY ANALOG SOURCE IDENTIFICATION PROBLEM – 3 POINTS NEAR 256 HZ

- **The three points that are closest to the band located within 10% of 256 Hz are 229.1 Hz, 251.2 Hz, and 275.4 Hz.**
  - If the voltage equals the ideal value of 1/2 volts in this interval, the deviation is 0.0.
  - If the voltage is within 240 millivolts of 1/2 volts, the absolute value of the deviation from 1/2 volts is weighted by a factor of 20.
  - If the voltage is more than 240 millivolts from 1/2 volts, the absolute value of the deviation from 1/2 volts is weighted by a factor of 200.
- **This arrangement reflects the fact that the ideal output voltage for this range of frequencies is 1/2 volts, that a 240 millivolts discrepancy is acceptable, and that a larger discrepancy is not acceptable.**

FITNESS MEASURE – 3 POINTS NEAR 2,560 HZ

- **The three points that are closest to the band located within 10% of 2,560 Hz are 2,291 Hz, 2,512 Hz, and 2,754 Hz.**
  - If the voltage equals the ideal value of 1 volt in this interval, the deviation is 0.0.
  - If the voltage is within 240 millivolts of 1 volt, the absolute value of the deviation from 1 volt is weighted by a factor of 20.
  - If the voltage is more than 240 millivolts from 1 volt, the absolute value of the deviation from 1 volt is weighted by a factor of 200.

FITNESS MEASURE FOR THREE-WAY ANALOG SOURCE IDENTIFICATION PROBLEM – REMAINING 95 POINTS IN THE FREQUENCY DOMAIN

- **The procedure for each of the remaining 95 points is as follows:**
  - If the voltage equals the ideal value of 0 volts, the deviation is 0.0.
  - If the voltage is within 240 millivolts of 0 volts, the absolute value of the deviation from 0 volts is weighted by a factor of 1.0.
  - If the voltage is more than 240 millivolts from 0 volts, the absolute value of the deviation from 0 volt is weighted by a factor of 10.
- **Greater weights (20 and 200) were used in the two passbands because they contain only 6 of the 101 points.**

FITNESS MEASURE FOR THREE-WAY ANALOG SOURCE IDENTIFICATION PROBLEM – CONTINUED

- **Many of the circuits that are created in the initial random population and many that are created by the crossover and mutation operations cannot be simulated by SPICE. Such circuits are assigned a high penalty value of fitness (10^8).**

- **The number of hits is defined as the number of fitness cases (0 to 101) for which the voltage is acceptable or ideal.**

CONTROL PARAMETERS FOR THREE-WAY ANALOG SOURCE IDENTIFICATION PROBLEM

- **Population size, *M*, of 640,000**

- **Maximum number of generations, *G*, is set to be meaninglessly large**

- **Maximum of H_{rpb} = 600 points (functions and terminals) for the result-producing branch**

- **For each generation**
  - 10% reproductions
  - 1% mutations
  - 89% crossovers
  - No architecture-altering operations

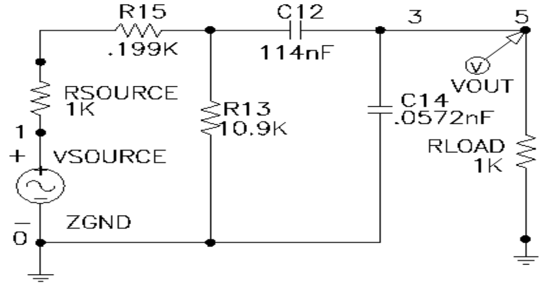
- **Secondary parameters are default values in Koza 1994 (appendix D)**

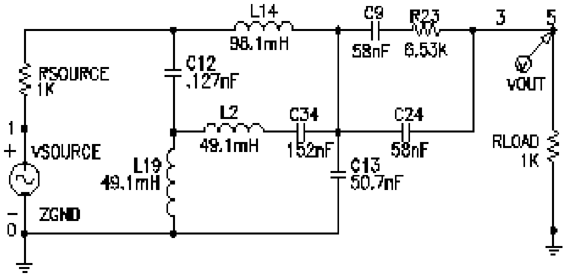
TERMINATION CRITERION AND RESULTS DESIGNATION FOR THREE-WAY ANALOG SOURCE IDENTIFICATION PROBLEM

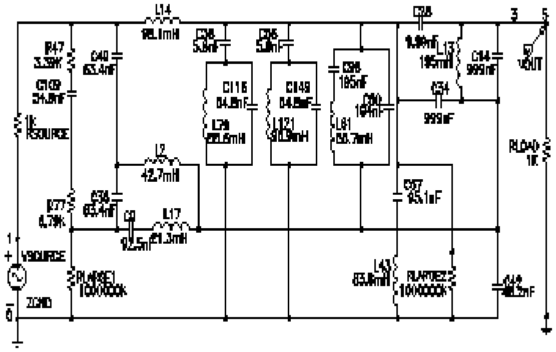
- **Manual intervention in lieu of pre-established termination criterion**

- **Best-so-far individual is designated as the result of the run**

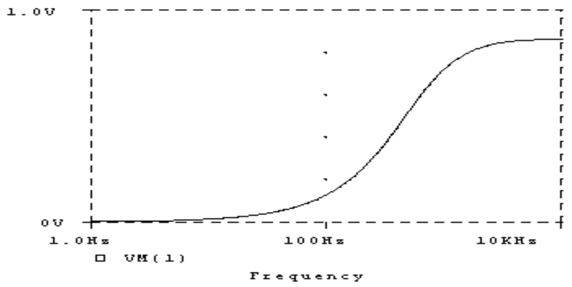
RESULTS FOR THE THREE-WAY ANALOG SOURCE IDENTIFICATION PROBLEM – GENERATIONS 0, 20, 106

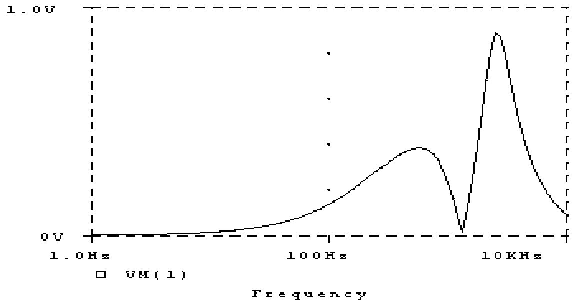


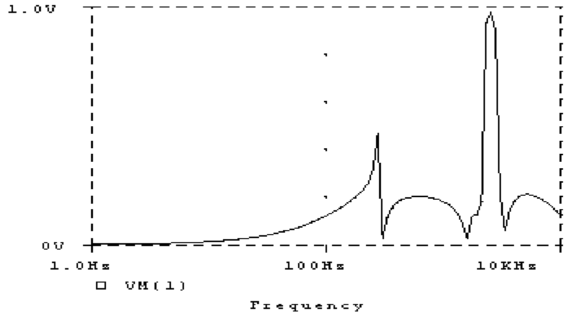




RESULTS FOR THE THREE-WAY ANALOG SOURCE IDENTIFICATION PROBLEM – GENERATIONS 0, 20, 106







THE CHANGING ENVIRONMENT PROBLEM (WITH ADFs AND ARCHITECTURE-ALTERING OPERATIONS)

- The goal is to evolve the design for a circuit that changes its structure as the number of different sources increases.
  - Initially the circuit classifies the incoming signals into **three** categories
  - Later the circuit undergoes modification so that is can successfully classify them into **four** categories.

THE CHANGING ENVIRONMENT PROBLEM (WITH ADFs AND ARCHITECTURE-ALTERING OPERATIONS) – PHASE 1 (3-WAY)

- **During phase 1, the requirements for the desired circuit are similar to those for the tri-state frequency discriminator except that one of the desired outputs is 1/3 volt (instead of 1/2 volt).**
  - The desired circuit is to produce an output of 1/3 volts (plus or minus 166 millivolts) if the frequency of the incoming signal is within 10% of 256 Hz
  - produce an output of 1 volt (plus or minus 166 millivolts) if the frequency of the incoming signal is within 10% of 2,560 Hz, and
  - otherwise produce an output of 0 volts (plus or minus 166 millivolts).

THE CHANGING ENVIRONMENT PROBLEM (WITH ADFs AND ARCHITECTURE-ALTERING OPERATIONS) – PHASE 2 (4-WAY)

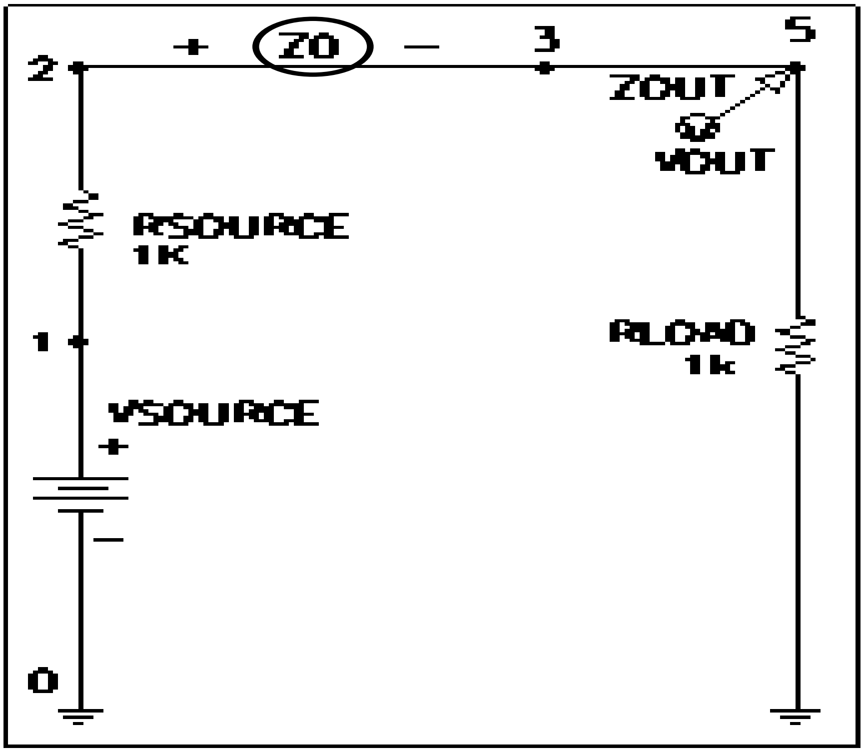
- **After a circuit is evolved that performs the tri-state source identification task, the requirements of the problem are changed to include an additional frequency band.**

- **The run is continued with the existing population until a new circuit is evolved that performs the source identification task for all three frequency bands.**
  - During phase 2, the circuit is to produce an output of 2/3 volts (plus or minus 166 millivolts) if the frequency of the incoming signal is within 10% of 750 Hz
  - While still producing an output of 1/3, 1, and 0 volts (plus or minus 166 millivolts) for the original three signals.

PREPARATORY STEPS FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFs AND ARCHITECTURE-ALTERING OPERATIONS)

ONE-INPUT, ONE-OUTPUT EMBRYO WITH ONE WRITING HEAD (ONE MODIFIABLE WIRE)

SAME AS BEFORE



ARCHITECTURE OF CIRCUIT-CONSTRUCTING PROGRAM TREE FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFs AND ARCHITECTURE-ALTERING OPERATIONS)

- **Since the initial circuit has one modifiable wire (and hence one writing head), there is one result-producing branch in each circuit-constructing program tree.**

- **Each program in the initial population of programs has a uniform architecture with no automatically defined functions. The number of automatically defined functions, if any, will emerge as a consequence of the evolutionary process using the architecture-altering operations.**

FUNCTION AND TERMINAL SETS OF RESULT PRODUCING BRANCH FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFS AND ARCHITECTURE-ALTERING OPERATIONS)

- **The initial function set, Fccs-initial, for each construction-continuing subtree is**

$F_{ccs\text{-}rpb}$ = {`R`, `L`, `C`, `SERIES`, `PARALLEL0`,
`PARALLEL1`, `FLIP`, `NOP`,
`T_PAIR_CONNECT_0`,
`T_PAIR_CONNECT_1`}

- **The initial terminal set, Tccs-initial, for each construction-continuing subtree is**

Tccs-initial = {`END`, `SAFE_CUT`}.

- **The set of potential new functions, Fpotential, is**

Fpotential = {`ADF0`, `ADF1`, `ADF2`}.

- **The set of potential new terminals, Tpotential, is**

Tpotential = {`ARG0`}.

FUNCTION AND TERMINAL SETS OF RESULT PRODUCING BRANCH FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFs AND ARCHITECTURE-ALTERING OPERATIONS) – CONTINUED

- **The architecture-altering operations change the function set, F_{ccs} for each construction-continuing subtree of both the result-producing branches and the function-defining branches, so**

$F_{ccs} = F_{ccs\text{-initial}} \approx F_{potential}$.

- **The architecture-altering operations change the terminal set, $T_{aps\text{-}adf}$, for each arithmetic-performing subtree, so**

$T_{aps\text{-}adf} = T_{aps\text{-}initial} \approx T_{potential}$.

FUNCTION AND TERMINAL SETS FOR ARITHMETIC-PERFORMING SUBTREES (USED IN BOTH RBP AND ADFs)

SAME AS BEFORE

- **The function set, Faps, for each arithmetic-performing subtree is,**

Faps = {+, -}.

- **The terminal set, Taps, for each arithmetic-performing subtree consists of**

Taps = {$\Re$},

where $\Re$ represents floating-point random constants from –1.0 to +1.0.

FITNESS MEASURE FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFS AND ARCHITECTURE-ALTERING OPERATIONS) – PHASE 1 (3-WAY)

- **During the first phase, there are only two frequencies of interest (256 Hz and 2,560 Hz); however, in the second phase, there are three frequencies of interest (750 Hz in addition the two just mentioned).**

FITNESS MEASURE FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFS AND ARCHITECTURE-ALTERING OPERATIONS) – PHASE 1 (3-WAY) – POINTS NEAR 256 HZ

- **The three points that are closest to the band located within 10% of 256 Hz are 229.1 Hz, 251.2 Hz, and 275.4 Hz.**
  - If the voltage equals the ideal value of 1/3 volts in this interval, the deviation is 0.0.

  - If the voltage is more than 166 millivolts from 1/3 volts, the absolute value of the deviation from 1/3 volts is weighted by a factor of 20.
  - If the voltage is more than 166 millivolts of 1/3 volts, the absolute value of the deviation from 1/3 volts is weighted by a factor of 200.

- **This arrangement reflects the fact that the ideal output voltage for this range of frequencies is 1/3 volts, that a 166 millivolts discrepancy is acceptable, and that a larger discrepancy is not acceptable.**

FITNESS MEASURE – PHASE 1 (3-WAY) – POINTS NEAR 2,560 HZ

- **The three points that are closest to the band located within 10% of 2,560 Hz are 2,291 Hz, 2,512 Hz, and 2,754 Hz.**
  - If the voltage equals the ideal value of 1 volt in this interval, the deviation is 0.0.
  - If the voltage is within 166 millivolts of 1 volt, the absolute value of the deviation from 1 volt is weighted by a factor of 20.
  - If the voltage is more than 166 millivolts from 1 volt, the absolute value of the deviation from 1 volt is weighted by a factor of 200.

FITNESS MEASURE – PHASE 1 (3-WAY) – REMAINING 95 POINTS

- **The procedure for each of the remaining 95 points is as follows:**
    - If the voltage equals the ideal value of 0 volts, the deviation is 0.0.
    - If the voltage is within 166 millivolts of 0 volts, the absolute value of the deviation from 0 volts is weighted by a factor of 1.0.
    - If the voltage is more than 166 millivolts from 0 volts, the absolute value of the deviation from 0 volt is weighted by a factor of 10.
- **Greater weights (20 and 200) were used in the two passbands because they contain only 6 of the 101 points.**

FITNESS MEASURE FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFs AND ARCHITECTURE-ALTERING OPERATIONS) – PHASE 2 (4-WAY) – POINTS NEAR 750 HZ

- **In phase 2, frequencies around 750 Hz come into play.**

- **The three points that are closest to the band located within 10% of 750 Hz are 791.8 Hz, 758.6 Hz, and 831.8 Hz.**
  - If the voltage equals the ideal value of 2/3 volts in this interval, the deviation is 0.0.
  - If the voltage is more than 166 millivolts from 2/3 volts, the absolute value of the deviation from 2/3 volts is weighted by a factor of 15.
  - If the voltage is more than 166 mV of 2/3 volts, the absolute value of the deviation from 2/3 volts is weighted by 150.

FITNESS MEASURE FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFS AND ARCHITECTURE-ALTERING OPERATIONS) – PHASE 2 (4-WAY) – POINTS NEAR 256 HZ AND 2,560 HZ

- **In phase 2, the procedure for the six points nearest 256 Hz and 2,560 Hz are the same as before, except that**
  - the weight is 15 and 150 (instead of 20 and 200), respectively for the complaint and non-complaint points.
  - Lesser weights (15 and 150) were used in the three passbands because 9 of the 101 points lie in the passbands.

FITNESS MEASURE FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFS AND ARCHITECTURE-ALTERING OPERATIONS) – PHASE 2 (4-WAY) – REMAINING 92 POINTS

- **In phase 1, the procedure for each of the remaining 92 points is as follows:**
  - If the voltage equals the ideal value of 0 volts, the deviation is 0.0.
  - If the voltage is within 166 millivolts of 0 volts, the absolute value of the deviation from 0 volts is weighted by a factor of 1.0.
  - If the voltage is more than 166 mV from 0 volts, the absolute value of the deviation from 0 is weighted by a factor of 10.
- **As before, for each phase, the number of hits is defined as the number of fitness cases for which the voltage is acceptable or ideal.**

CONTROL PARAMETERS FOR THE CHANGING ENVIRONMENT PROBLEM

- **The percentage of operations on each generation after generation 5 was**
  - 86.5% one-offspring crossovers;
  - 10% reproductions;
  - 1% mutations;
  - 1% branch duplications;
  - 0.5% branch deletions;
  - 1% branch creations; and
  - 0% argument creations.
- **The percentage of operations on each generation before generation 6 was**
  - 78.0% one-offspring crossovers;
  - 10% reproductions;
  - 1% mutations;
  - 5.0% branch duplications;
  - 1% branch deletions;
  - 5.0% branch creations; and
  - 0% argument creations.

CONTROL PARAMETERS FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFs AND ARCHITECTURE-ALTERING OPERATIONS) – CONTINUED

- **The maximum size, H_{rpb}, for the result-producing branch is 600 points.**

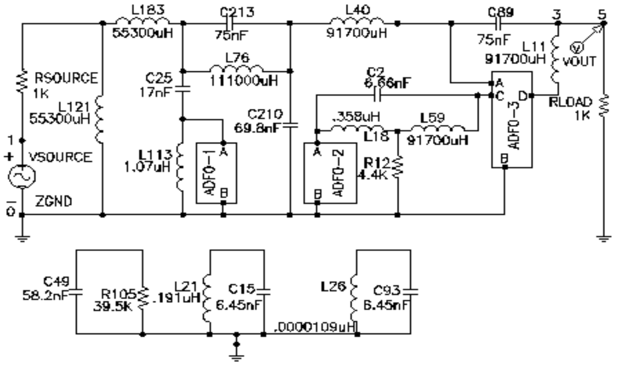
- **The maximum number of automatically defined functions is 2.**

- **The number of arguments for each automatically defined function is 1.**

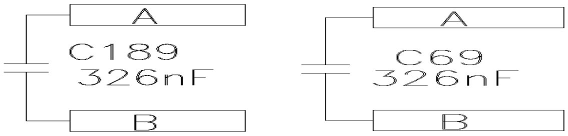
- **The maximum size, H_{adf}, for each of the automatically defined functions, if any, is 300 points.**

RESULTS FOR THE CHANGING ENVIRONMENT PROBLEM

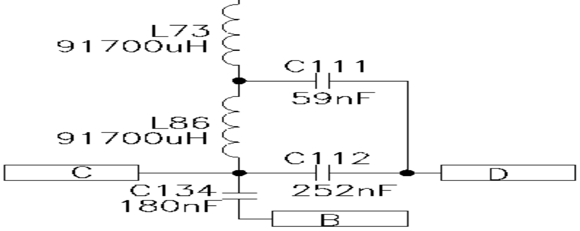
BEST CIRCUIT FROM GENERATION 41 BEFORE EXPANSION OF THE THREE OCCURRENCES OF `ADF0`



RESULT OF DEVELOPING `ADF0-1` AND `ADF0-1` FOR BEST CIRCUIT FROM GENERATION 41

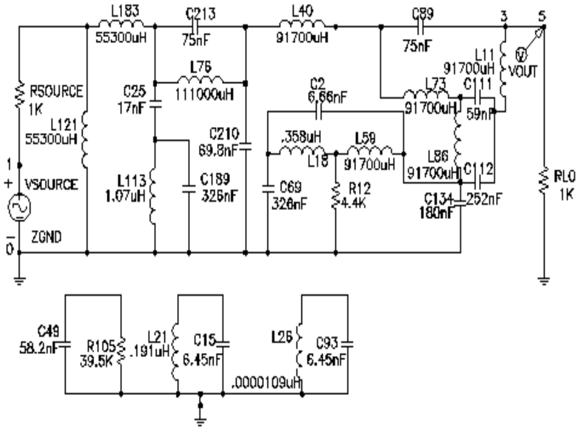


RESULT OF DEVELOPING `ADF0` FOR BEST CIRCUIT FROM GENERATION 41



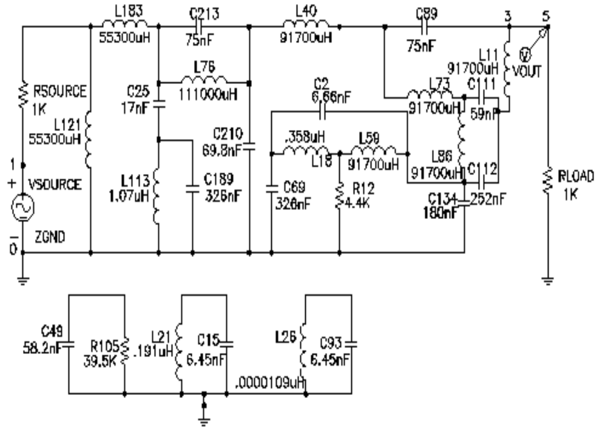
RESULTS FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFs AND ARCHITECTURE-ALTERING OPERATIONS)

BEST CIRCUIT FROM GENERATION 41 AFTER EXPANSION OF THE THREE OCCURRENCES OF `ADF0`



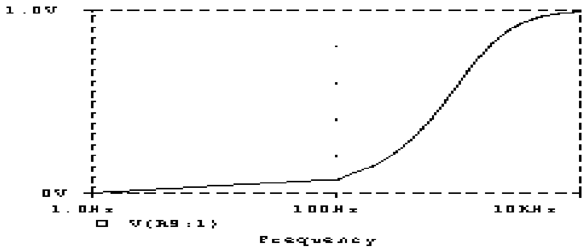
RESULTS FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFs AND ARCHITECTURE-ALTERING OPERATIONS)

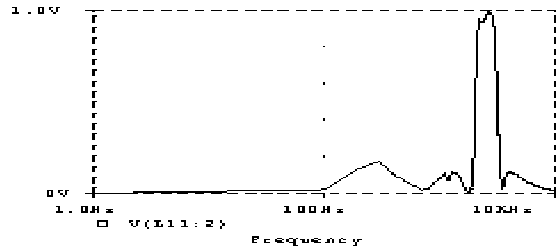
BEST CIRCUIT FROM GENERATION 85 AFTER EXPANSION OF ITS AUTOMATICALLY DEFINED FUNCTIONS

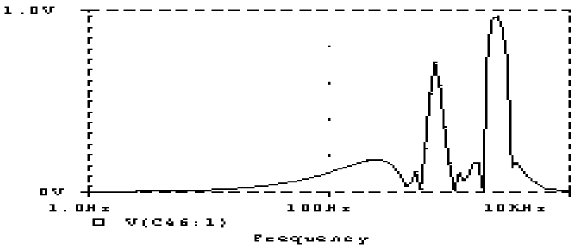


RESULTS FOR THE CHANGING ENVIRONMENT PROBLEM (WITH ADFs AND ARCHITECTURE-ALTERING OPERATIONS)

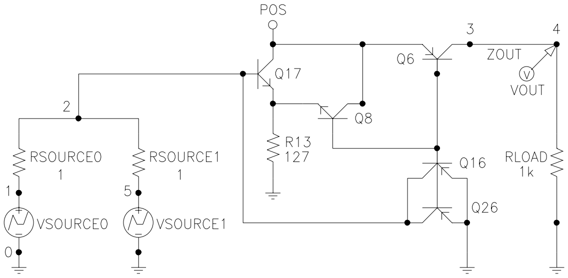
FREQUENCY DOMAIN BEHAVIOR OF THE BEST CIRCUIT OF GENERATION 0, 41, AND 85



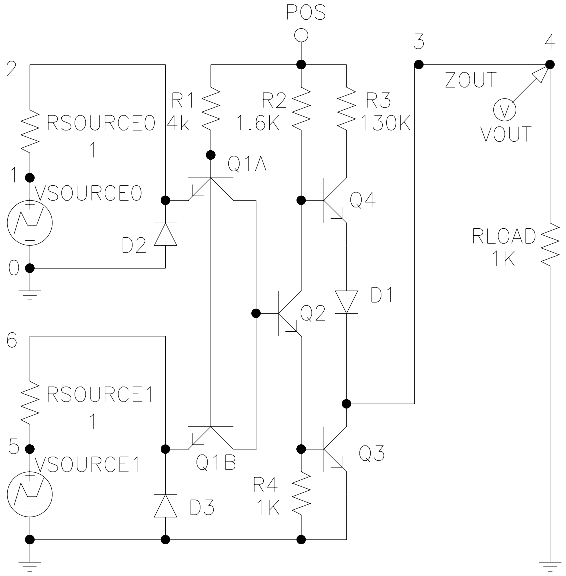




EVOLVED 100 NANO-SECOND NAND GATE



TEXTBOOK NAND GATE

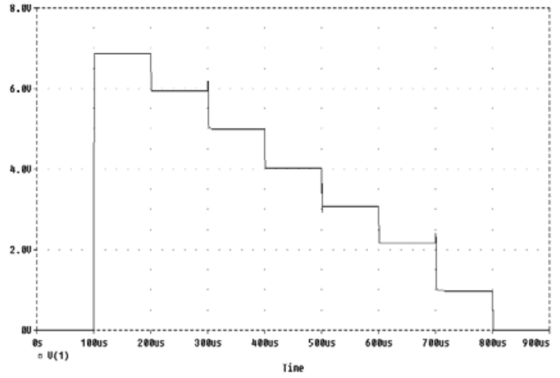


FIRST FITNESS MEASURE

- **Sum, over the 92 fitness cases, of the weighted absolute value of the difference between the actual output voltage and the desired output voltage (0 to 7 volts).**
  - If the voltage exactly equals the desired voltage of VSOURCE0 plus two times VSOURCE1 plus four times VSOURCE2, the deviation is 0.
  - If the voltage is within 0.25 volts of the desired voltage, the absolute value of the deviation from the desired output voltage is weighted by a factor of 1.0.
  - If the voltage is outside this range, the absolute value of the deviation is weighted by a factor of 10.0.

100 NANO-SECOND DIGITAL-TO-ANALOG CONVERTED (DAC)

GLITCH-RIDDEN BEHAVIOR OF BEST-OF-RUN EVOLVED DAC USING THE FIRST FITNESS MEASURE



100 NANO-SECOND DIGITAL-TO-ANALOG CONVERTED (DAC)

SECOND FITNESS MEASURE

- **Our second fitness measure was the sum, for each of SPICE's internally created "turns," of the areas of the trapezoids between the curve representing the desired output voltage and the curve representing the actual output voltage.**
- **The magnitude of the glitches in the best-of-run circuit that was evolved using this second fitness measure were as large as 1 volt — far larger that the magnitude of the glitches that we were trying to eliminate.**
- **The second fitness measure tolerated these larger glitches because they were very narrow (and hence occupied very little total area).**

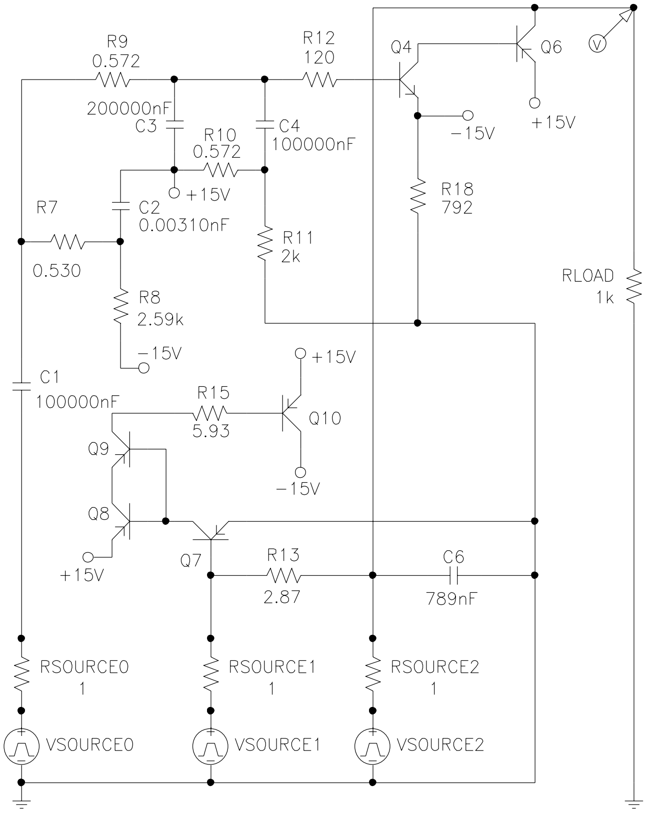
100 NANO-SECOND DIGITAL-TO-ANALOG CONVERTED (DAC)

THIRD FITNESS MEASURE

- **"Crossover" of Our First Two Fitness Measures combined some of the features of the first and second fitness measures.**
- **The third fitness measure was the sum, over SPICE's turn-defining points, of the weighted absolute value of the difference between the actual output voltage and the desired output voltage.**
  - If the voltage exactly equals the desired voltage of VSOURCE0 plus two times VSOURCE1 plus four times VSOURCE2, the deviation is 0.
  - If the voltage is within 0.25 volts of the desired voltage, the absolute value of the deviation from the desired output voltage is weighted by a factor of 1.0.
  - If the voltage is outside this range, the absolute value of the deviation is weighted by a factor of 10.0.

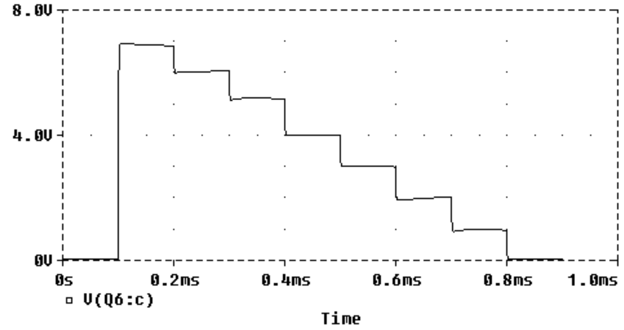
100 NANO-SECOND DIGITAL-TO-ANALOG CONVERTED (DAC)

BEST-OF-RUN EVOLVED CIRCUIT USING THE THIRD FITNESS MEASURE (GENERATION 139)



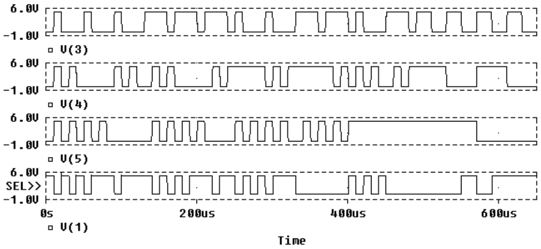
100 NANO-SECOND DIGITAL-TO-ANALOG CONVERTED (DAC)

BEHAVIOR OF BEST-OF-RUN EVOLVED DAC USING THE THIRD FITNESS MEASURE (GENERATION 139)



TWO-INSTRUCTION ALU CIRCUIT

64 10-μS DIGITAL SIGNALS — EACH SAMPLED EVERY 2 μS FOR A TOTAL OF 321 FITNESS CASES



TIME-OPTIMAL FLY-TO PROBLEM

USING GP TO EVOLVE AN EXPRESSION COMPOSED OF ARITHMETIC AND CONDITIONAL OPERATIONS

Terminal set for the one result-producing branch, T of a program in the population for the problem is

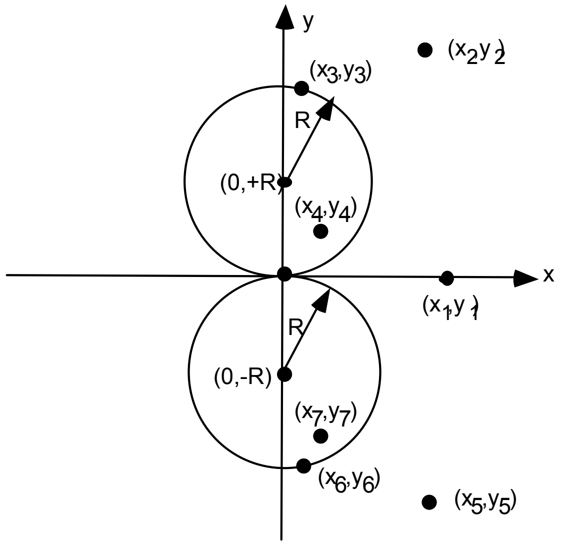
T = {X, Y}

The function set, F is

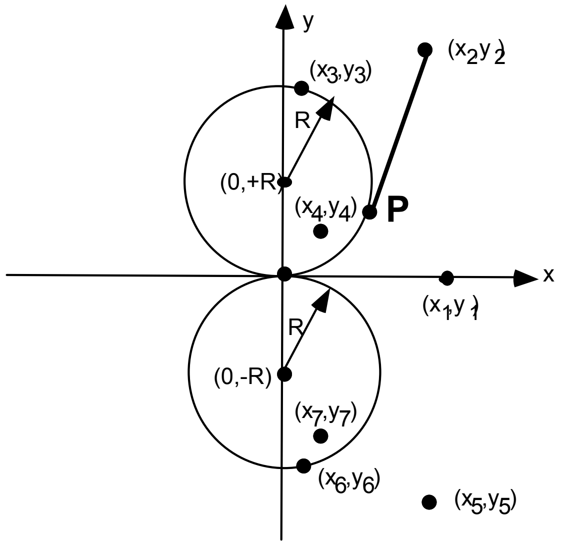
F= {+, –, *, %, IFLTE, $\Re$}

TIME-OPTIMAL FLY-TO PROBLEM

CASES 1 AND 2

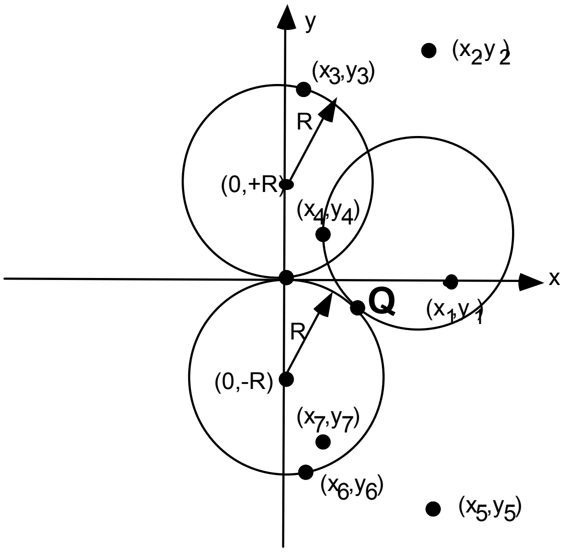


CASE 3



TIME-OPTIMAL FLY-TO PROBLEM

CASE 4



TIME-OPTIMAL FLY-TO PROBLEM USING GP TO EVOLVE AN ELECTRICAL CIRCUIT

FUNCTION AND TERMINAL SETS

Fccs = {`R`, `SERIES`, `PARALLEL0`, `PARALLEL1`, `FLIP`, `NOP`, `NEW_T_GND_0`, `NEW_T_GND_1`, `NEW_T_POS_0`, `NEW_T_POS_1`, `NEW_T_NEG_0`, `NEW_T_NEG_1`, `PAIR_CONNECT_0`, `PAIR_CONNECT_1`, `Q_D_NPN`, `Q_D_PNP`, `Q_3_NPN0`, ..., `Q_3_NPN11`, `Q_3_PNP0`, ..., `Q_3_PNP11`, `Q_POS_COLL_NPN`, `Q_GND_EMIT_NPN`, `Q_NEG_EMIT_NPN`, `Q_GND_EMIT_PNP`, `Q_POS_EMIT_PNP`, `Q_NEG_COLL_PNP`}

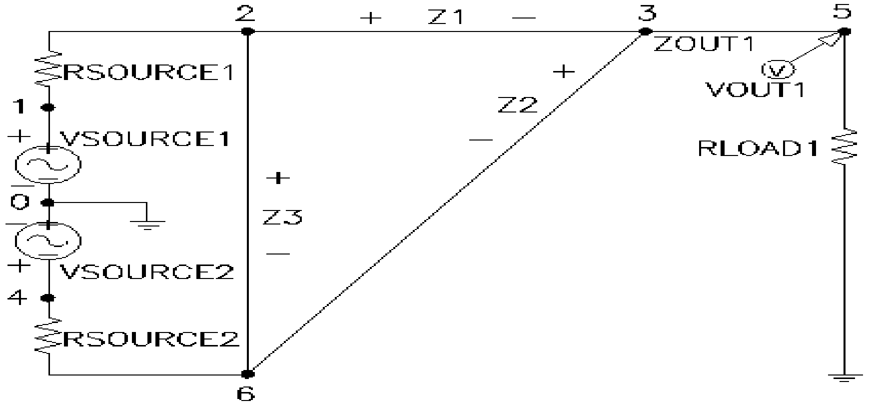
Tccs = {`END`, `SAFE_CUT`}

Faps = {`+`, `-`}

Taps = {$\Re$}

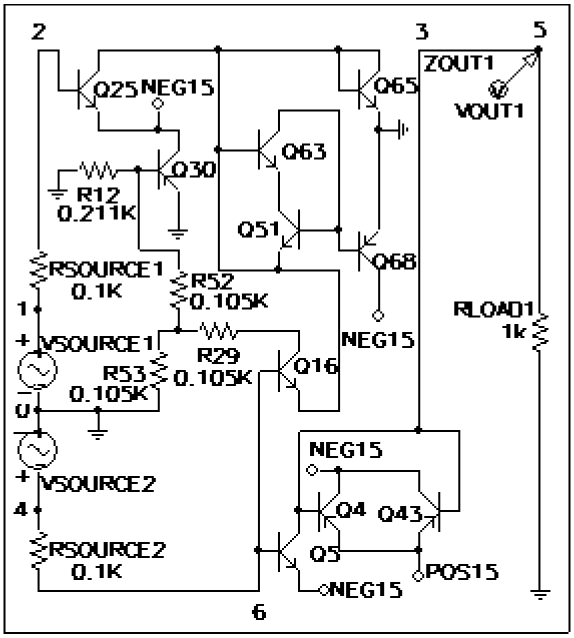
TIME-OPTIMAL "FLY-TO" PROBLEM USING GP TO EVOLVE AN ELECTRICAL CIRCUIT

TWO-INPUT, ONE-OUTPUT EMBRYO



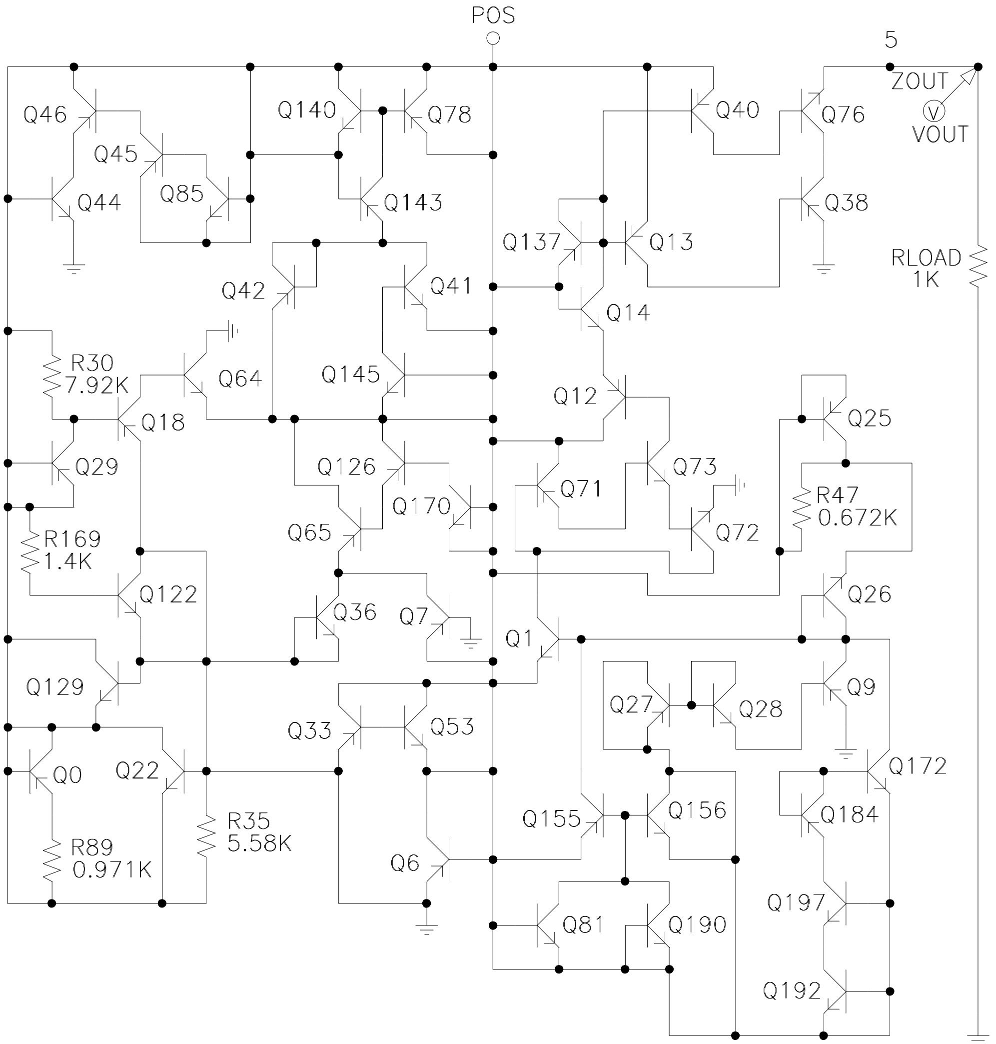
- **72 random destinations (x_i, y_i) in the plane**
- **Fitness is the sum, over the 72 destinations, of the <u>TIME</u> for the aircraft (robot) to reach the destination**

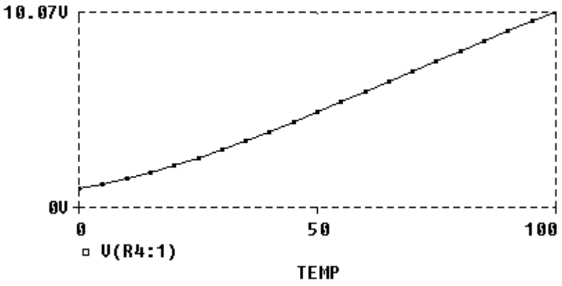
TIME-OPTIMAL FLY-TO PROBLEM – GENERATIONS 31 (WITH NEAR-OPTIMAL FITNESS SCORING 72 HITS OF 1.541 HOURS ON GENERATION 31)



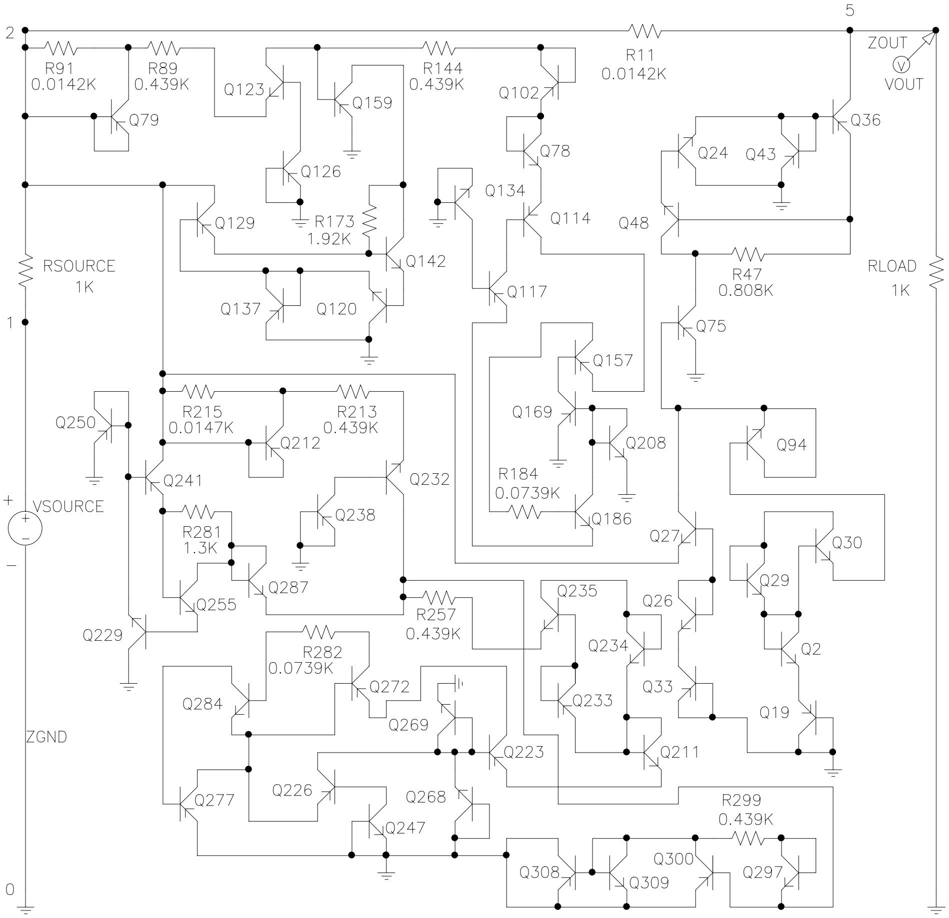
- **The circuit is automatically created using <u>TIME</u> as the fitness measure.**

ZERO-INPUT TEMPERATURE-SENSING CIRCUIT — GENERATION 25

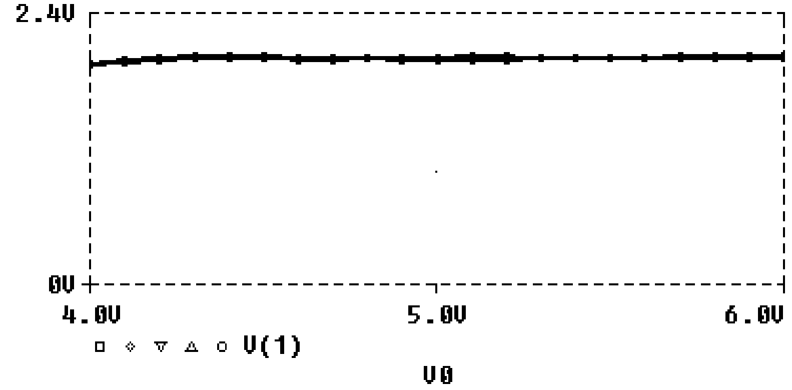




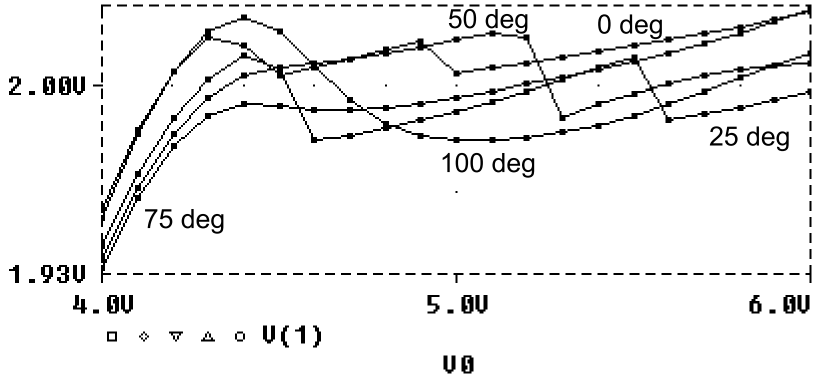
VOLTAGE REFERENCE CIRCUIT — GENERATION 80



VOLTAGE REFERENCE CIRCUIT — GENERATION 80

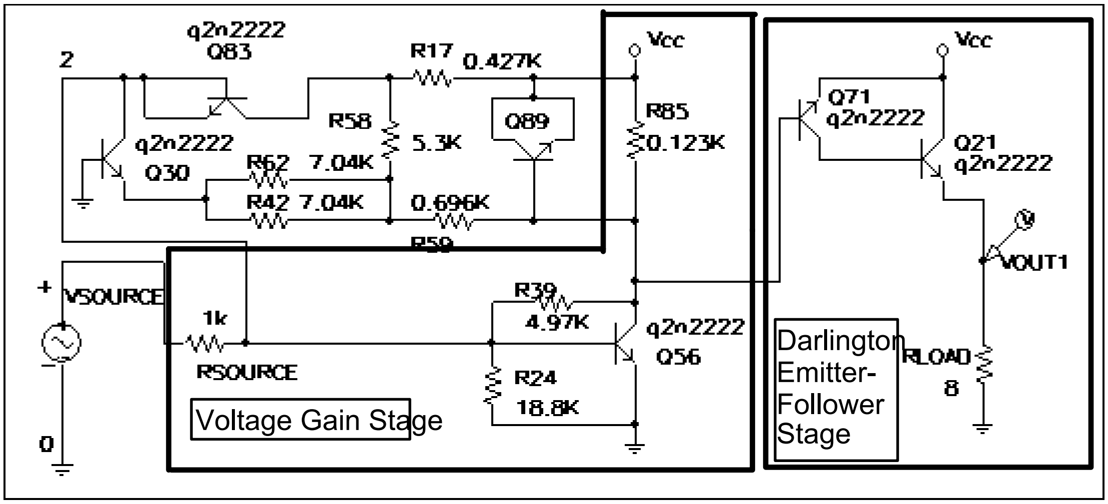


CLOSE-UP (NEAR 2 VOLTS)



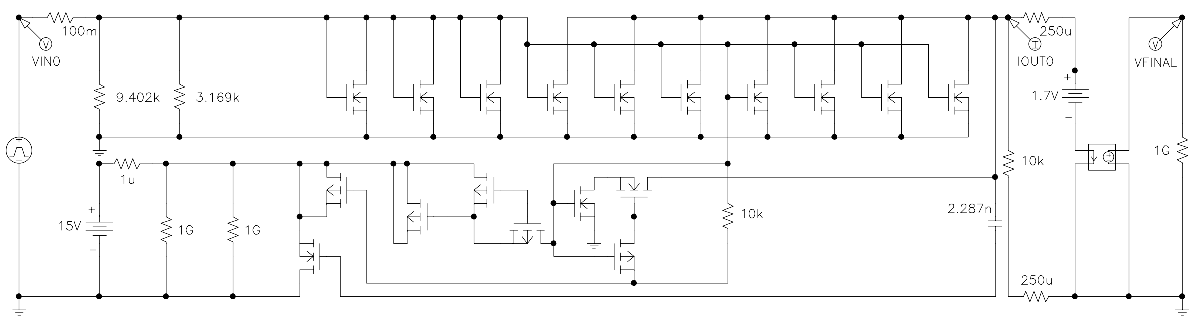
GENETICALLY EVOLVED 10 DB AMPLIFIER FROM GENERATION 45

SHOWING THE VOLTAGE GAIN STAGE AND DARLINGTON EMITTER FOLLOWER SECTION



POST-2000 PATENTED INVENTIONS

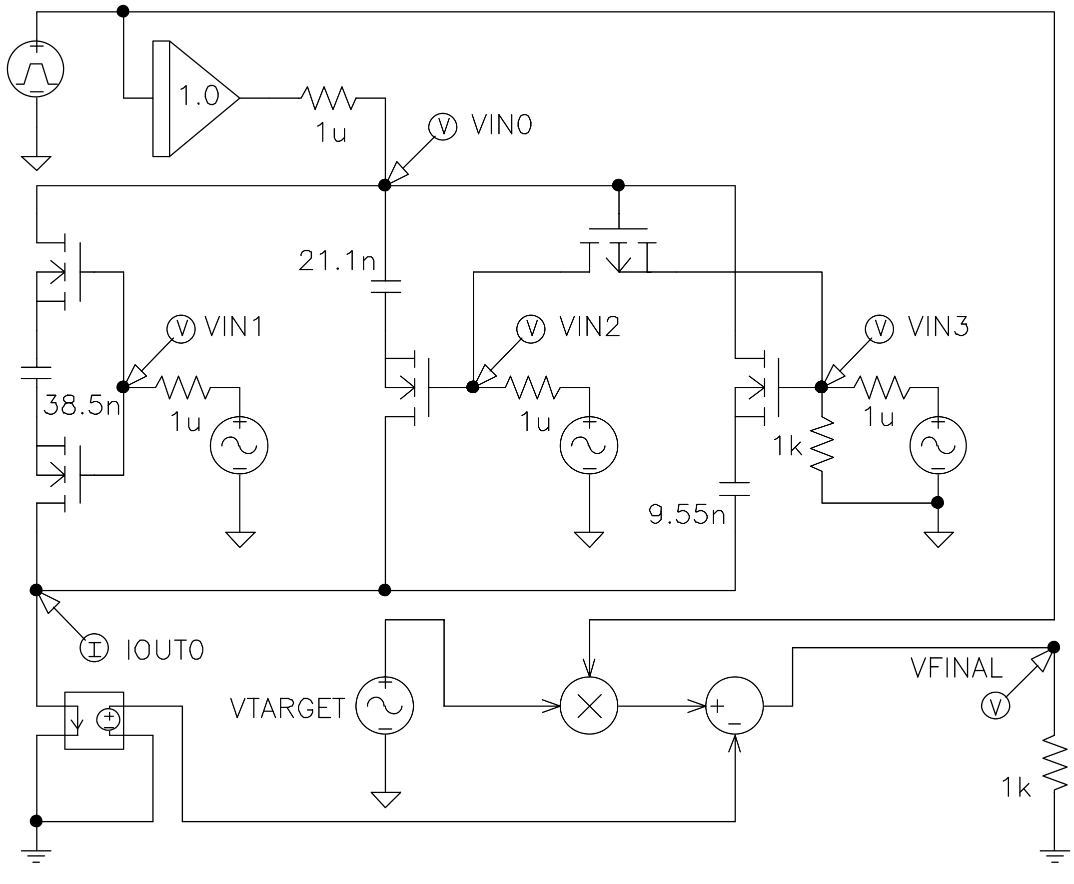
HIGH CURRENT LOAD CIRCUIT
BEST-OF-RUN FROM GENERATION 114



POST-2000 PATENTED INVENTIONS

REGISTER-CONTROLLED CAPACITOR CIRCUIT

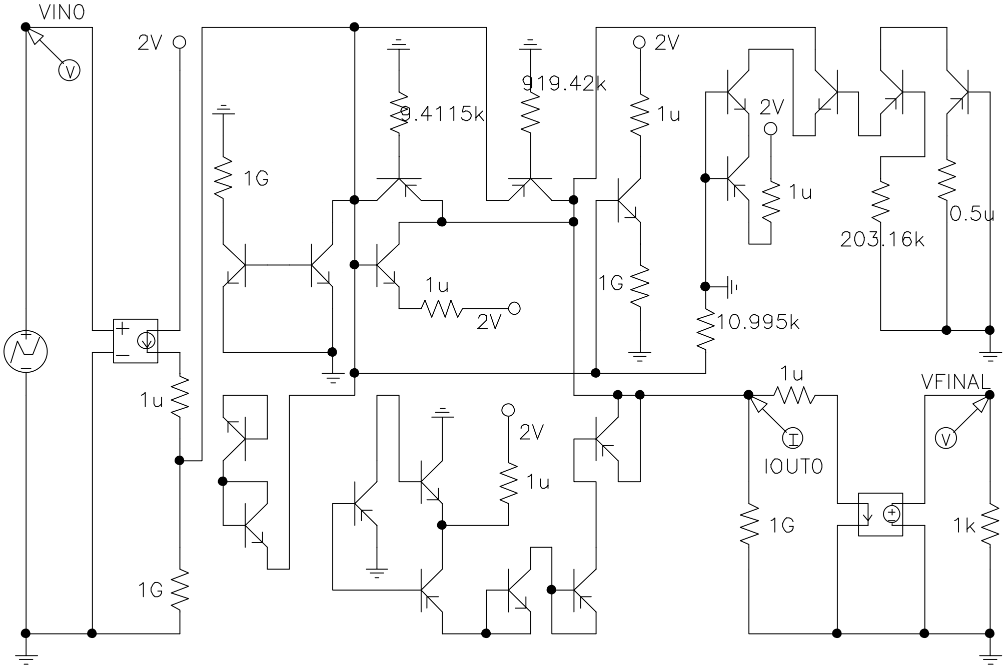
SMALLEST COMPLIANT FROM GENERATION 98



POST-2000 PATENTED INVENTIONS

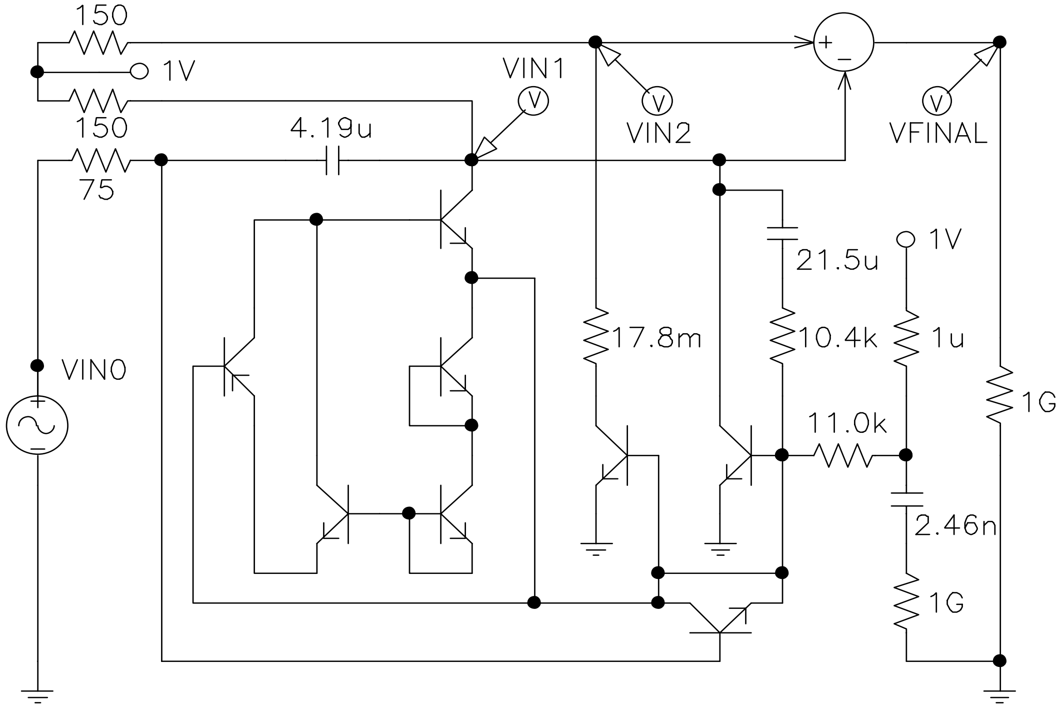
LOW-VOLTAGE CUBIC SIGNAL GENERATION CIRCUIT

BEST-OF-RUN FROM GENERATION 182



POST-2000 PATENTED INVENTIONS

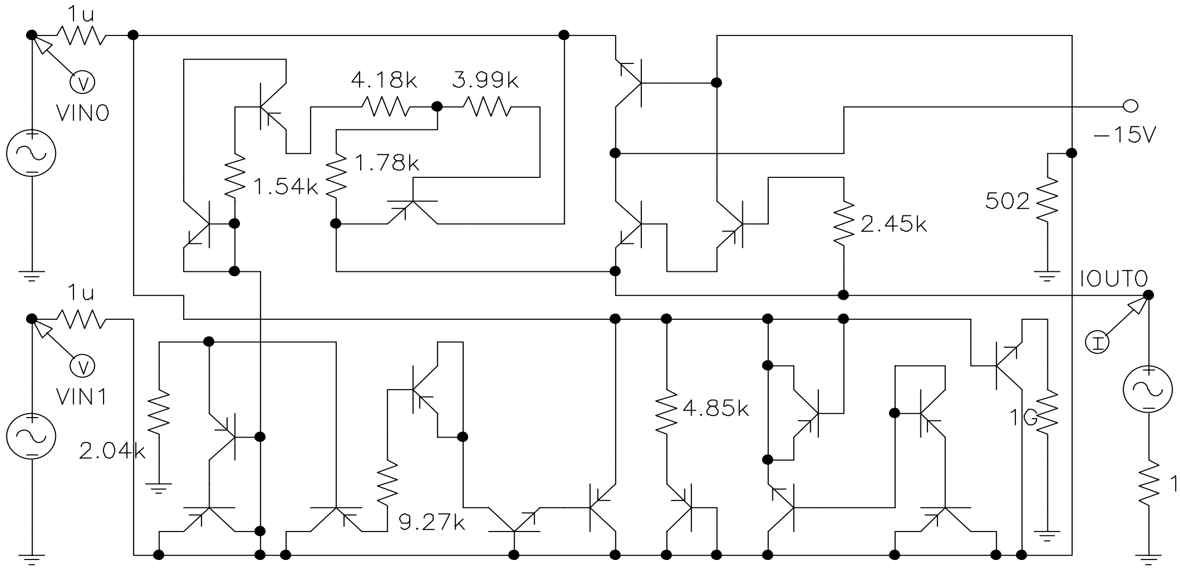
LOW-VOLTAGE BALUN CIRCUIT BEST EVOLVED FROM GENERATION 84



POST-2000 PATENTED INVENTIONS

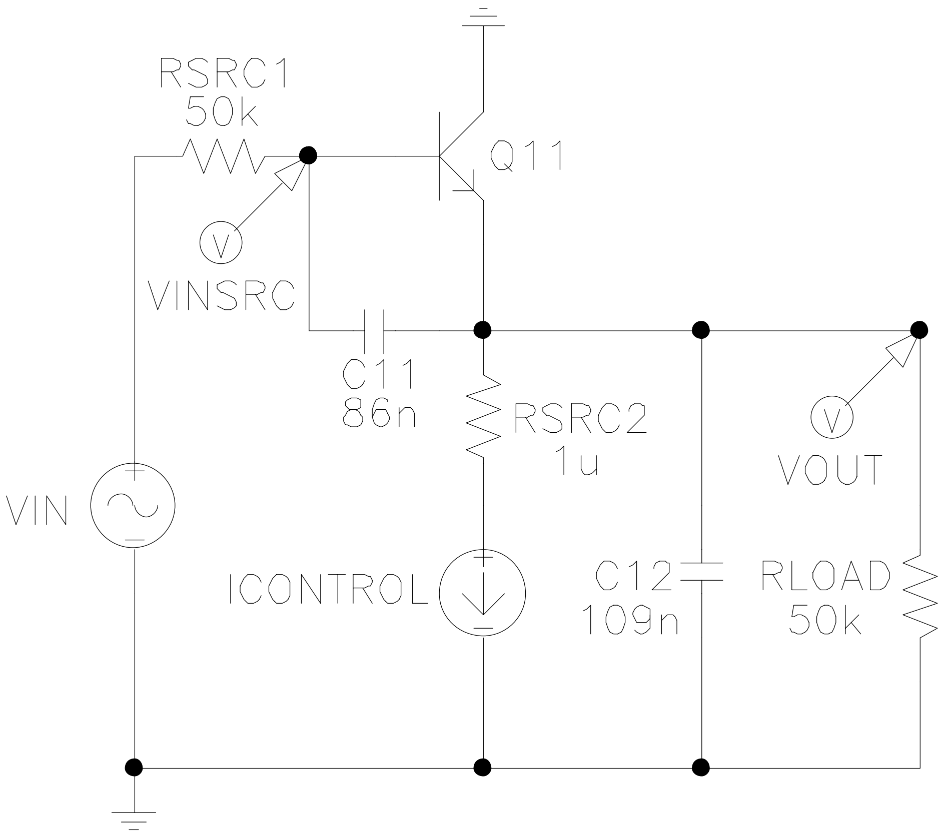
VOLTAGE-CURRENT-CONVERSION CIRCUIT

BEST-OF-RUN FROM GENERATION 109



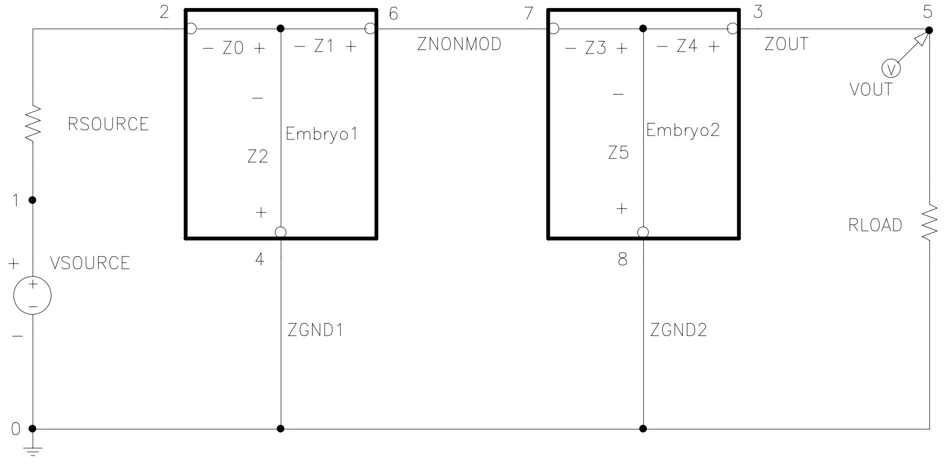
POST-2000 PATENTED INVENTIONS

TUNABLE INTEGRATED ACTIVE FILTER — GENERATION 50



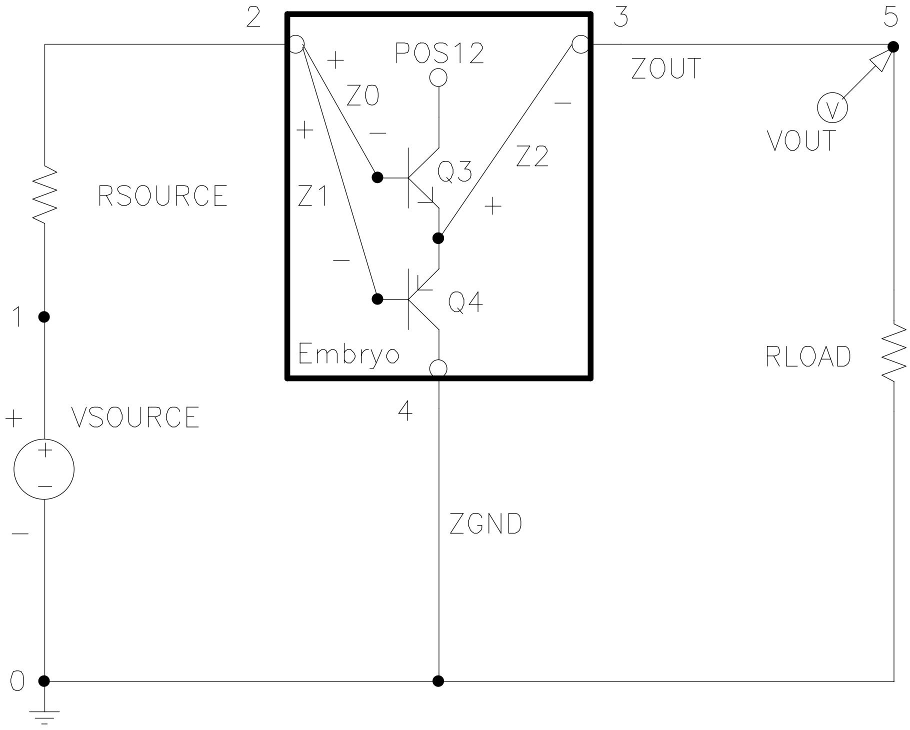
INCORPORATING TOPOLOGICAL CONSTRAINTS INTO DEVELOPMENTAL GP

2 SPECIAL EMBRYOS FOR A 2-STAGE CIRCUIT



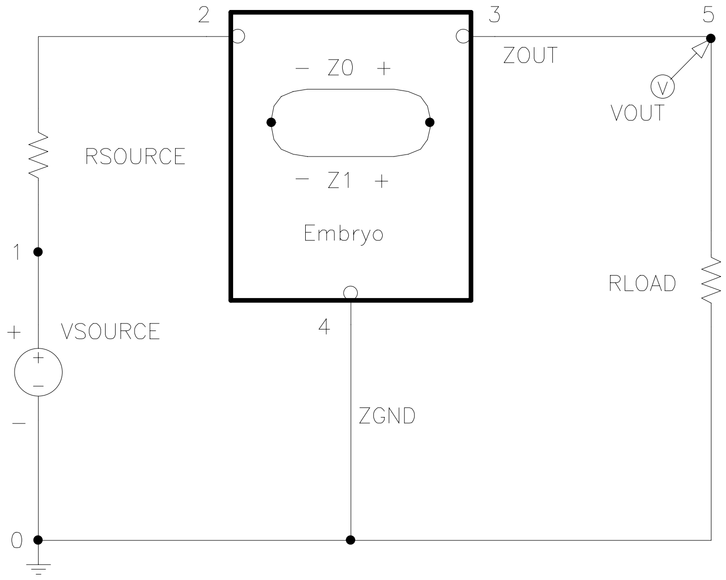
SUBCIRCUIT CONSTRAINTS

EMBRYO WITH TWO HARD-WIRED TRANSISTORS



MINIMAL EMBRYO

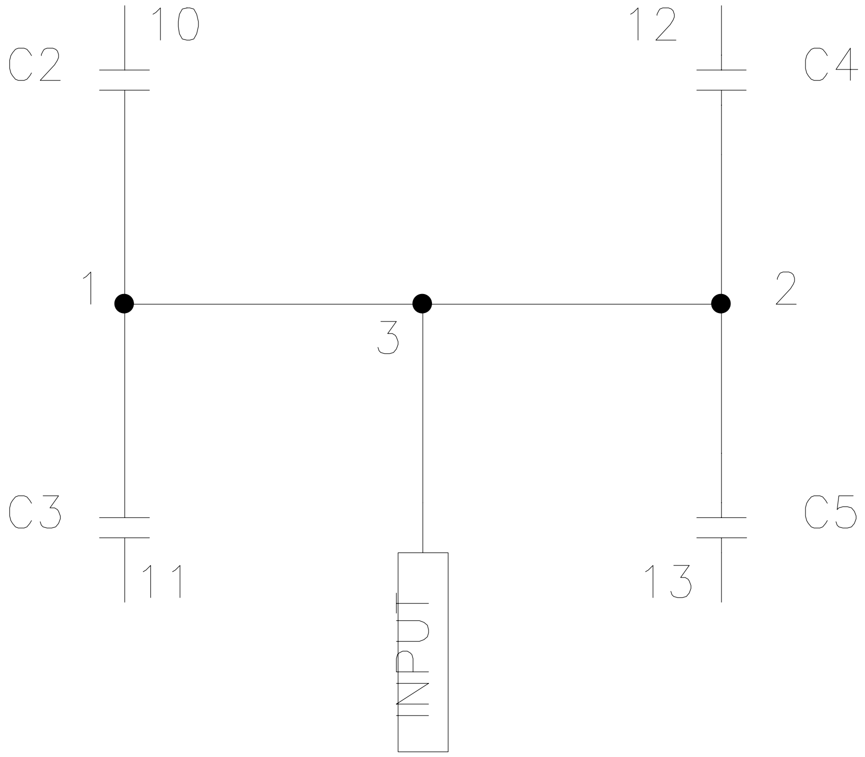
- **An embryo with two modifiable wires (`Z0` and `Z1`) and three ports in a one-input, one-output test fixture, but no direct access to input or output**



$T_{\text{ccs-rpb-initial}}$ = {`CUT`, `END`, `NOP`}

$F_{\text{ccs-rpb-initial}}$ = {`C`, `L`, `PARALLEL0`, `PARALLEL1`, `ZERO_GROUND`, `INPUT`, `OUTPUT`}

THE `INPUT` FUNCTION



100%-COMPLIANT BEST-OF-RUN CIRCUIT FROM GENERATION 92