EMERGENCE OF CREATIVITY (AND OTHER THINGS) IN GENETIC PROGRAMMING

SFI-UM-CSCS WORKSHOP
ANN ARBOR
THURSDAY NOVEMBER 13, 2003

John R. Koza
Consulting Professor (Medical Informatics)
Department of Medicine
School of Medicine
Stanford University

Consulting Professor
Department of Electrical Engineering
School of Engineering
Stanford University

E-MAIL: koza@stanford.edu
http://www.smi.stanford.edu/people/koza/

WORKSHOP DESCRIPTION

"the … challenge of creating … complex systems with particular structural or dynamic properties, e.g., to create:

- **architectural or computer chip designs that achieve desired goals subject to various constraints;**

…

- **physical robots to do specified jobs …**

WORKSHOP DESCRIPTION — CONTINUED

“Part of the challenge in creating complex systems like those listed is that <u>while some properties can be more-or-less directly designed into a system, other properties emerge</u> from the interaction of the system's parts.

“In addition, there is a growing interest in approaches to design that don't just take the emergent properties as complexities to be kept under control, but instead <u>try to harness those emergent properties to achieve design goals</u>”

GENETIC PROGRAMMING

WHAT'S EMERGENT IN GENETIC PROGRAMMING?

DEFINITION OF “EMERGENT”

DEFINITION OF “EMERGENT”

- **Getting more than you’re entitled to**

WHAT’S EMERGENT IN GENETIC PROGRAMMING?

- **Solutions to problems — Starting from a high-level statement of the problem**

36 HUMAN-COMPETITIVE RESULTS

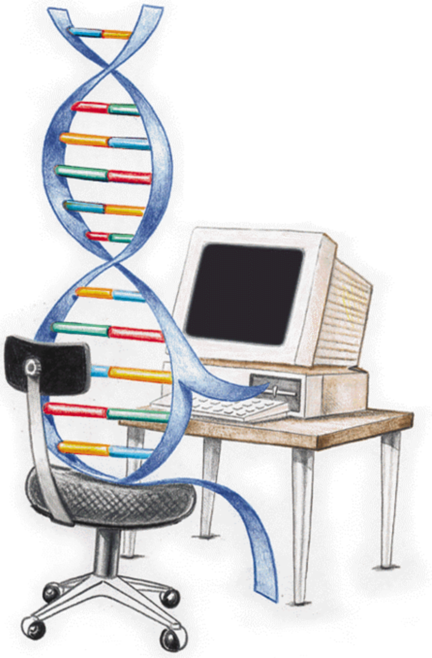
	Claimed instance	Basis for claim of human-competitiveness	Reference
1	Creation of a better-than-classical quantum algorithm for the Deutsch-Jozsa "early promise" problem	B, F	Spector, Barnum, and Bernstein 1998
2	Creation of a better-than-classical quantum algorithm for Grover's database search problem	B, F	Spector, Barnum, and Bernstein 1999
3	Creation of a quantum algorithm for the depth-two AND/OR query problem that is better than any previously published result	D	Spector, Barnum, Bernstein, and Swamy 1999; Barnum, Bernstein, and Spector 2000
4	Creation of a quantum algorithm for the depth-one OR query problem that is better than any previously published result	D	Barnum, Bernstein, and Spector 2000
5	Creation of a protocol for communicating information through a quantum gate that was previously thought not to permit such communication	D	Spector and Bernstein 2003
6	Creation of a novel variant of quantum dense coding	D	Spector and Bernstein 2003
7	Creation of a soccer-playing program that won its first two games in the Robo Cup 1997 competition	H	Luke 1998
8	Creation of a soccer-playing program that ranked in the middle of the field of 34 human-written programs in the Robo Cup 1998 competition	H	Andre and Teller 1999
9	Creation of four different algorithms for the transmembrane segment identification problem for proteins	B, E	Sections 18.8 and 18.10 of *GP-2 book* and sections 16.5 and 17.2 of GP-3 book
10	Creation of a sorting network for seven items using only 16 steps	A, D	Sections 21.4.4, 23.6, and 57.8.1 of GP-3 book
11	Rediscovery of the Campbell ladder topology for lowpass and highpass filters	A, F	Section 25.15.1 of GP-3 book and section 5.2 of GP-4 book
12	Rediscovery of the Zobel "*M*-derived half section" and "constant *K*" filter sections	A, F	Section 25.15.2 of GP-3 book
13	Rediscovery of the Cauer (elliptic) topology for filters	A, F	Section 27.3.7 of GP-3 book
14	Automatic decomposition of the problem of synthesizing a crossover filter	A, F	Section 32.3 of GP-3 book
15	Rediscovery of a recognizable voltage gain stage and a Darlington emitter-follower section of an amplifier and other circuits	A, F	Section 42.3 of GP-3 book
16	Synthesis of 60 and 96 decibel amplifiers	A, F	Section 45.3 of GP-3 book
17	Synthesis of analog computational circuits for squaring, cubing, square root, cube root, logarithm, and Gaussian functions	A, D, G	Section 47.5.3 of GP-3 book
18	Synthesis of a real-time analog circuit for time-optimal control of a robot	G	Section 48.3 of GP-3 book
19	Synthesis of an electronic thermometer	A, G	Section 49.3 of GP-3 book
20	Synthesis of a voltage reference circuit	A, G	Section 50.3 of GP-3 book

21	**Creation of a cellular automata rule for the majority classification problem that is better than the Gacs-Kurdyumov-Levin (GKL) rule and all other known rules written by humans**	**D, E**	**Andre, Bennett, and Koza 1996 and section 58.4 of GP-3 book**
22	**Creation of motifs that detect the D–E–A–D box family of proteins and the manganese superoxide dismutase family**	**C**	**Section 59.8 of GP-3 book**
23	**Synthesis of topology for a PID-D2 (proportional, integrative, derivative, and second derivative) controller**	**A, F**	**Section 3.7 of GP-4 book**
24	**Synthesis of an analog circuit equivalent to Philbrick circuit**	**A, F**	**Section 4.3 of GP-4 book**
25	**Synthesis of a `NAND` circuit**	**A, F**	**Section 4.4 of GP-4 book**
26	**Simultaneous synthesis of topology, sizing, placement, and routing of analog electrical circuits**	**A. F, G**	**Chapter 5 of GP-4 book**
27	**Synthesis of topology for a PID (proportional, integrative, and derivative) controller**	**A, F**	**Section 9.2 of GP-4 book**
28	**Rediscovery of negative feedback**	**A, E, F, G**	**Chapter 14 of GP-4 book**
29	**Synthesis of a low-voltage balun circuit**	**A**	**Section 15.4.1 of GP-4 book**
30	**Synthesis of a mixed analog-digital variable capacitor circuit**	**A**	**Section 15.4.2 of GP-4 book**
31	**Synthesis of a high-current load circuit**	**A**	**Section 15.4.3 of GP-4 book**
32	**Synthesis of a voltage-current conversion circuit**	**A**	**Section 15.4.4 of GP-4 book**
33	**Synthesis of a Cubic function generator**	**A**	**Section 15.4.5 of GP-4 book**
34	**Synthesis of a tunable integrated active filter**	**A**	**Section 15.4.6 of GP-4 book**
35	**Creation of PID tuning rules that outperform the Ziegler-Nichols and Åström-Hägglund tuning rules**	**A, B, D, E, F, G**	**Chapter 12 of GP-4 book**
36	**Creation of three non-PID controllers that outperform a PID controller that uses the Ziegler-Nichols or Åström-Hägglund tuning rules**	**A, B, D, E, F, G**	**Chapter 13 of GP-4 book**

WHAT’S EMERGENT IN GENETIC PROGRAMMING? — CONTINUED

- **Solutions to problems**
- **Reuse**
- **Hierarchy**
- **Architecture**
- **Creativity**

CREATIVITY



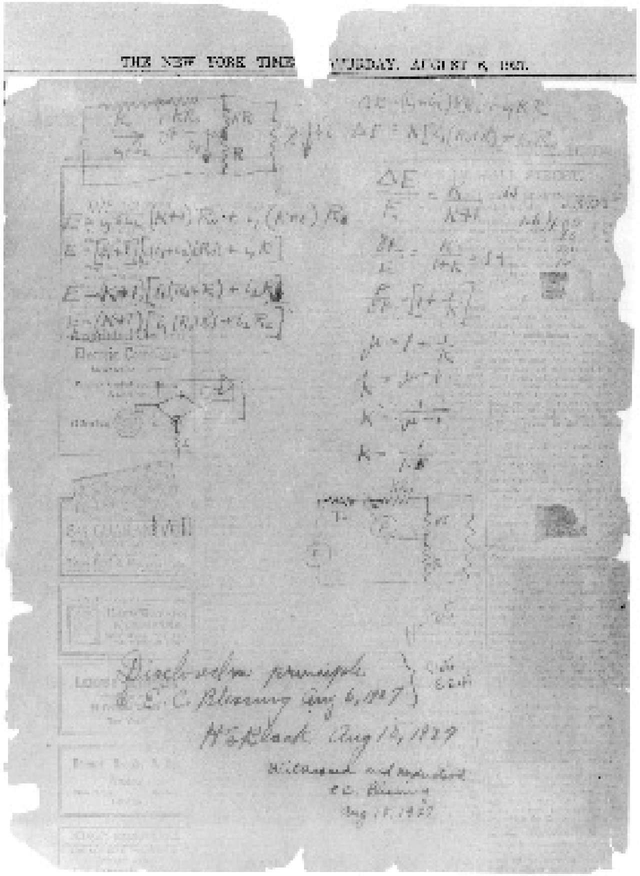
CREATIVITY

- **An illogical discontinuity**

HAROLD S. BLACK DESCRIPTION OF HIS RIDE ON THE LACKAWANNA FERRY

“Then came the morning of Tuesday, August 2, 1927, when the concept of the negative feedback amplifier *came to me in a flash* while I was crossing the Hudson River on the Lackawanna Ferry, on my way to work. For more than 50 years, I have pondered how and why the idea came, and I can’t say any more today than I could that morning. All I know is that after several years of hard work on the problem, I suddenly realized that if I fed the amplifier output back to the input …

NOTES WRITTEN BY HAROLD S. BLACK ON A PAGE OF *THE NEW YORK TIMES* WHILE COMMUTING ON THE LACKAWANNA FERRY



Disclosed this principle
to E. C. Blessing Aug 6, 1927

H S Black Aug 14, 1927

Witnessed and understood
E. C. Blessing
Aug 14, 1927

BLACK (1977)

“…more than nine years would elapse before the patent was issued … One reason for the delay was that *the concept was so contrary to established beliefs.*”

“… our patent application was treated in the same manner as one for a perpetual motion machine.”

ARMSTRONG'S POSITIVE TYPE OF FEEDBACK (1914)

"[P]rogress in electronics in those early years was largely made possible by Armstrong's regenerative [positive feedback] amplifier, since there was no other economical way to obtain large amounts of gain from the primitive (and expensive) vacuum tubes of the day."…

"In short order, *the positive feedback (regenerative) amplifier became a nearly universal idiom*,"

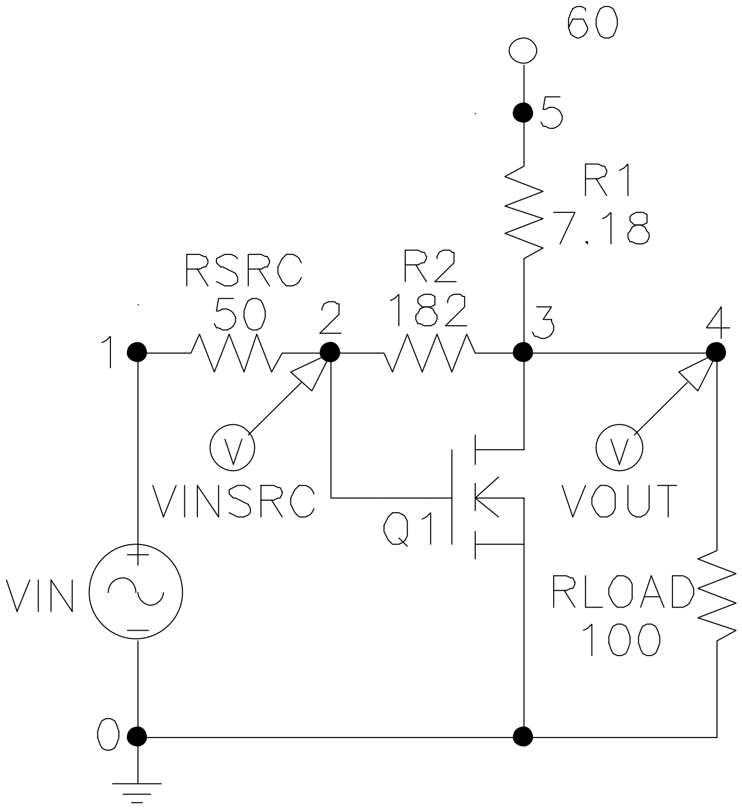
ARMSTRONG’S POSITIVE TYPE OF FEEDBACK (1914) — CONTINUED

- **Amplified incoming signal**
- **Distortion when things are non-linear**
- **Distortion overwhelms signal after repeated amplification**

BLACK'S WORK AT AT&T 1921-1927 ON PROBLEM OF REDUCING DISTORTION IN AMPLIFIERS

- **Black's first solution (1923) was entirely feed-forward**
- **The incoming signal is amplified by amplifier no. 1**
- **The amplified system is then cut back (by a resistive network) by the amplifier's amplification factor**
- **The cut-back (and usually somewhat distorted) signal is subtracted from the incoming signal to produce an error**
- **This error is then amplified by amplifier no. 2 (identical to no. 1) and then subtracted from the output of amplifier no. 1**
- **The result has little or no distortion**
- **This approach relies on identical amplifiers**
- **Impractical**

NEGATIVE FEEDBACK AMPLIFIER (USING `IRFZ44` FET TRANSISTOR IN LIEU OF VACUUM TUBES)



WHY WAS THE DISCOVERY OF NEGATIVE FEEDBACK SO DIFFICULT?

- **One reason why it took an inordinate amount of time for negative feedback to gain acceptance was that human thinking often becomes channeled along the well-traveled paths of "established beliefs."**

GEDANKEN EXPERIMENT USING A KNOWLEDGE-BASED LOGICAL APPROACH

- **Codify all the knowledge about electronics of the time (1927)**
- **There is no way to reach the idea of negative feedback by logic applied to the knowledge of the times**

GEDANKEN EXPERIMENT — CONTINUED

"… the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would [not] have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains."

— 35 *United States Code* 103a

GEDANKEN EXPERIMENT — CONTINUED

- **What’s missing is the “flash of genius — that is, the logical discontinuity**

GEDANKEN EXPERIMENT — CONTINUED

- **The human contribution is the *illogic***

BLACK'S SOLUTION READILY "EMERGES" FROM THE HIGH-LEVEL STATEMENT OF THE PROBLEM OF REDUCING DISOTORTION IN AMPLIFIERS

3-PART FITNESS MEASURE

- Amplification of the incoming signal
- Minimizing of distortion

$$THD = \frac{\sqrt{\sum_{i=2}^{N} A_i^2}}{A_1}$$

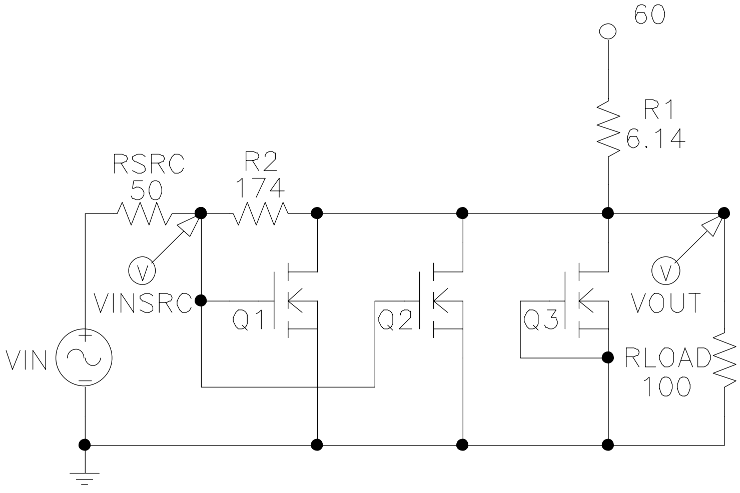
- Parsimony (number of components)

FUNCTIONS AND TERMINALS

- Allow creation of circuits with resistors, capacitors, and transistors (using a developmental process)

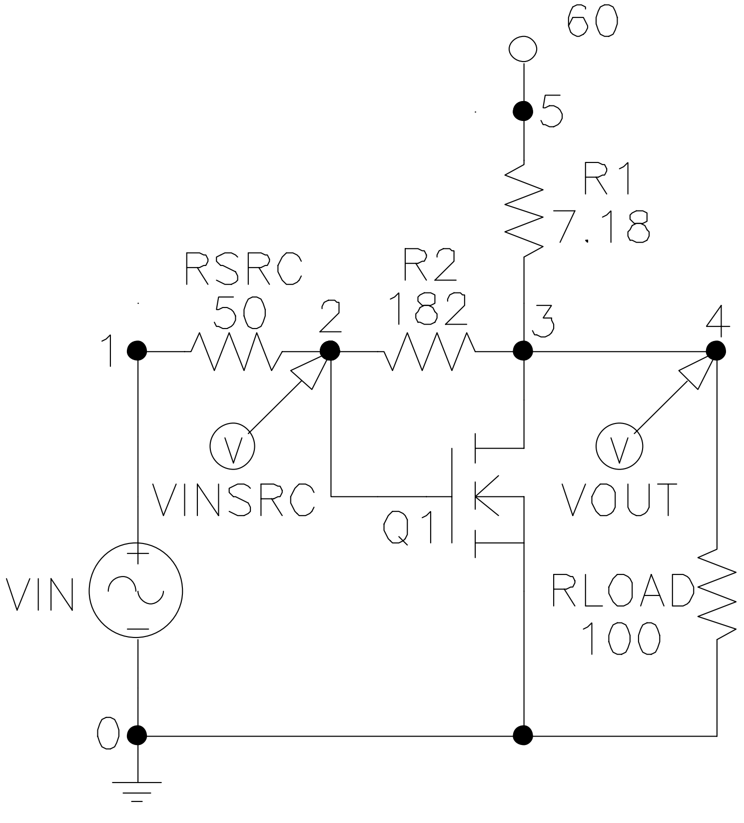
PROGRESS OF THE GP RUN

- The best circuit from generation 0 delivers amplification of –2.91 dB (i.e., acts as an attenuator rather than an amplifier)
- The first best-of-generation circuit that acts as an amplifier appeared in generation 9 (with only 5.37 dB amplification and total harmonic distortion of –5.65 dB)
- The first circuit satisfying the 10 dB amplification criterion and having a total harmonic distortion of less than –45 dB appeared in generation 46. This circuit has a total harmonic distortion of –54.2 dB.



BEST OF GENERATION 48

- Amplification of 10.06 dB
- Total harmonic distortion of –51.2 dB
- Infringes claims 1 and 3 of U.S. patent 2,102,671 (Black 1937)



BLACK (1977)

"Then came the morning of Tuesday, August 2, 1927, <u>when the concept of the negative feedback amplifier came to me in a flash</u> …

"… I suddenly realized that if I fed the amplifier output back to the input, in reverse phase, and kept the device from oscillating (singing, as we called it then), I would have exactly what I wanted: a means of canceling out the distortion of the output. I opened my morning newspaper and on a page of *The New York Times* I sketched a simple canonical diagram of a negative feedback amplifier plus the equations for the amplification with feedback. I signed the sketch, and 20 minutes later, when I reached the laboratory at 463 West Street, it was witnessed, understood, and signed by the late Earl C. Blessing."

THE AI RATIO

- **What is delivered by the actual automated operation of an artificial method in comparison to the amount of knowledge, information, analysis, and intelligence that is pre-supplied by the human employing the method?**

THE AI RATIO — CONTINUED

- **We define the *AI ratio* (the "artificial-to-intelligence" ratio) of a problem-solving method as the ratio of that which is delivered by the automated operation of the *artificial* method to the amount of *intelligence* that is supplied by the human applying the method to a particular problem.**

THE AI RATIO — CONTINUED

- **Deep Blue**
- **Chinook**

FROM JAWS-1 …

“STRUCTURE ARISES FROM FITNESS”

BLACK'S PROBLEM OF REDUCING DISTORTION IN AMPLIFIERS

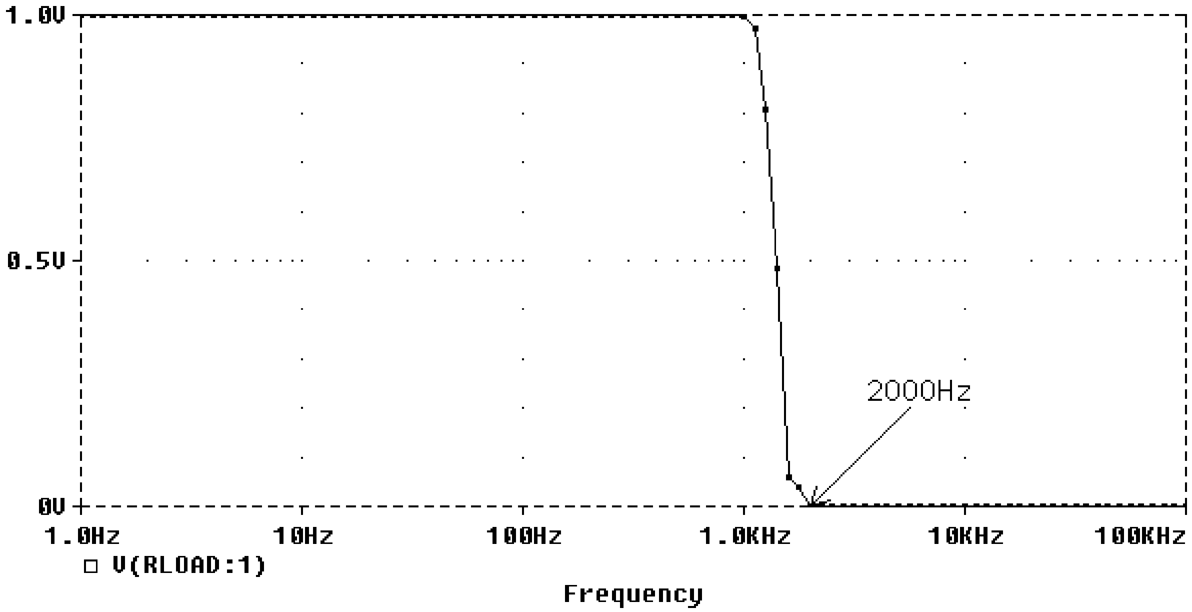
3-PART FITNESS MEASURE

- **amplifies the incoming signal**
- **minimizes distortion**

$$THD = \frac{\sqrt{\sum_{i=2}^{N} A_i^2}}{A_1}$$

- **contains a small number of components**

AUTOMATIC SYNTHESIS OF A LOWPASS FILTER



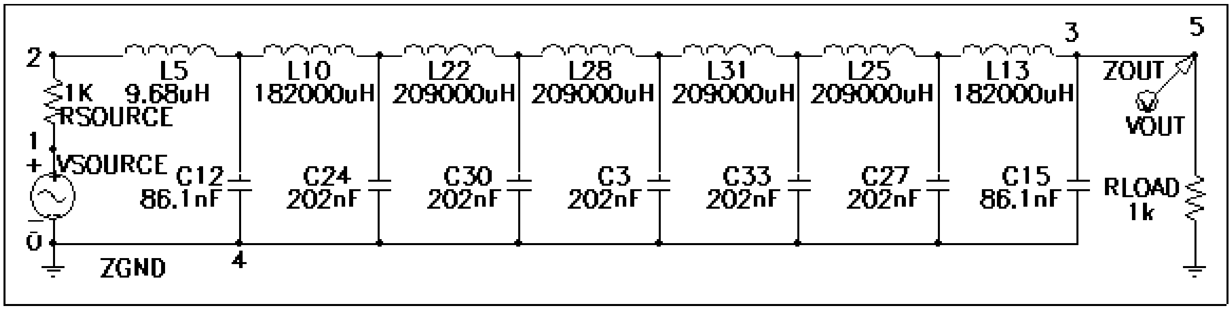
FITNESS MEASURE

- **Ideally 1 volt below 1,000 Hertz, but no worse than 970 millivolts**
- **Ideally 0 volts above 2,000 Hertz, but no higher than 1 millivolts**

FUNCTIONS AND TERMINALS

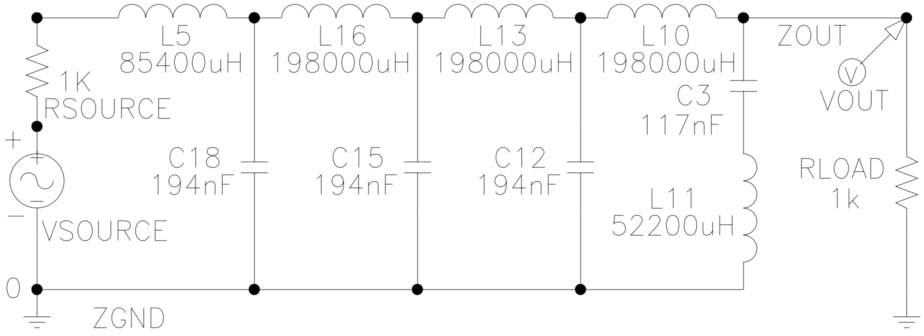
- **Allow creation of circuits with resistors, capacitors, and inductors (using a developmental process)**

EVOLVED CAMPBELL FILTER
(U. S. PATENT 1,227,113 — 1917)



- **Cascade of 6 symmetric π-sections**

EVOLVED ZOBEL FILTER (U. S. PATENT 1,538,964 — 1925)



- **Cascade of 3 symmetric T-sections**
- **One *M*-derived half section (`C2` and `L11`)**

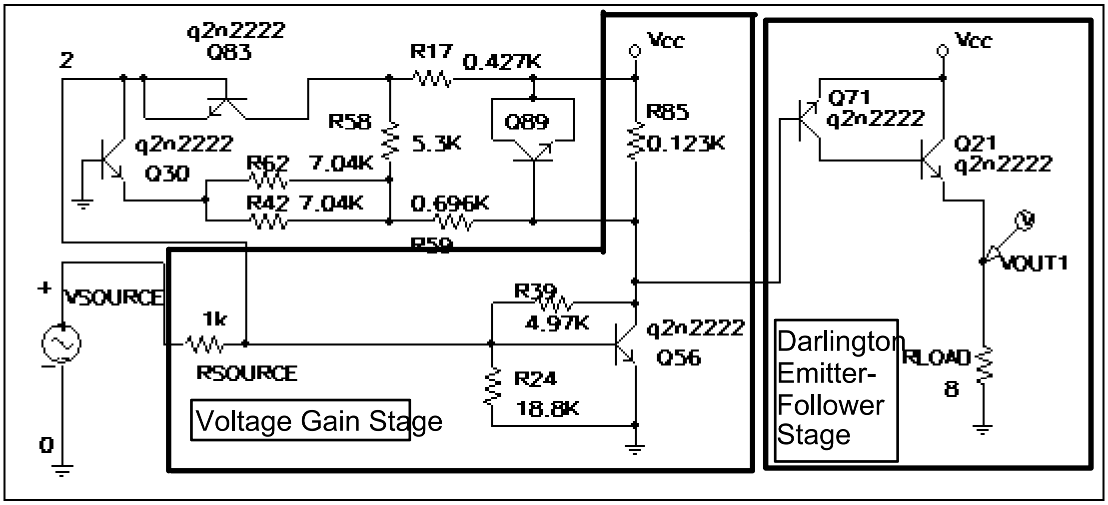
THE AI RATIO — CONTINUED

FROM JAWS-1 …

“STRUCTURE ARISES FROM FITNESS”

GENETICALLY EVOLVED 10 DB AMPLIFIER FROM GENERATION 45

SHOWING A VOLTAGE GAIN STAGE AND QUASI-DARLINGTON EMITTER FOLLOWER SECTION



DARLINGTON EMITTER-FOLLOWER SECTION'(U. S. PATENT 2,663,806 — 1953)

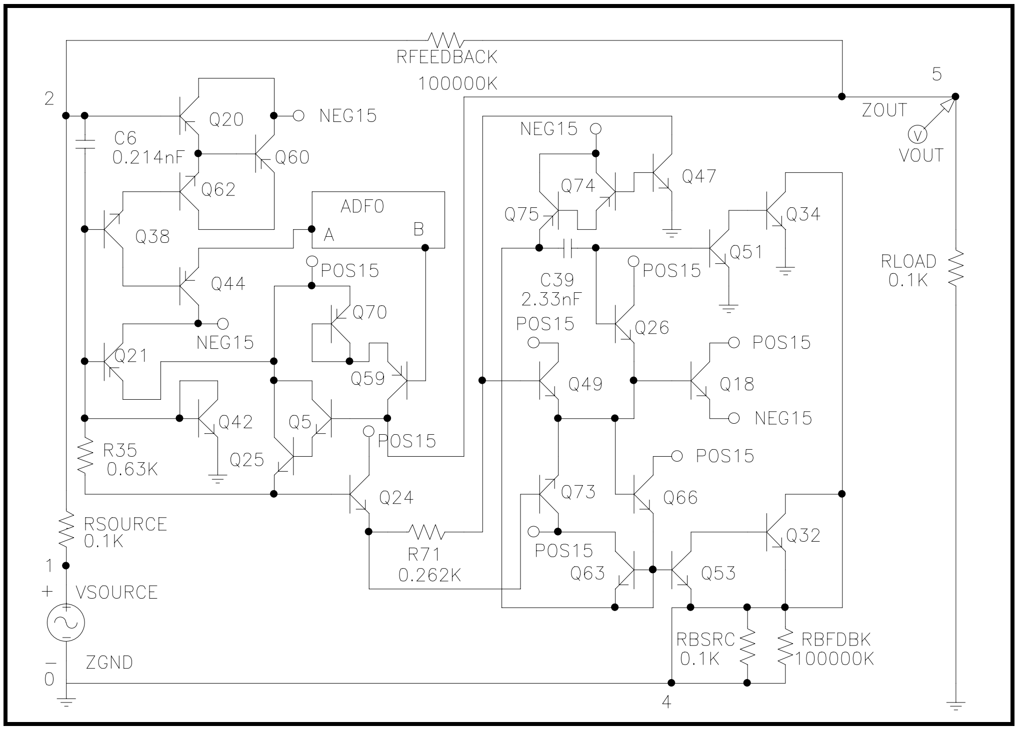
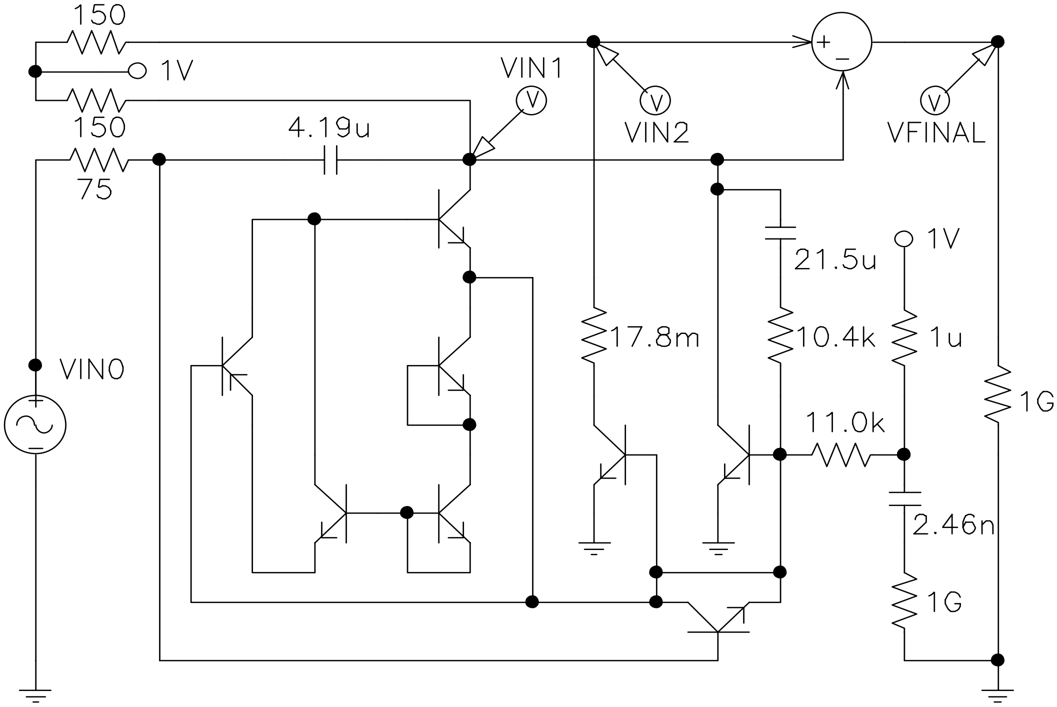


Figure I n GP-3 book	Circuit	Transistors	Patent claim
45.16	96 dB amplifier	Q5 and Q25	1
45.16	96 dB amplifier	Q53, Q32	3
47.6	Squaring computational	Q101, Q119	1
47.6	Squaring computational	Q29, Q88	4
47.10	Cubing computational	Q27, Q46	3
47.10	Cubing computational	Q46, Q35	3
47.11	Cubing computational	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

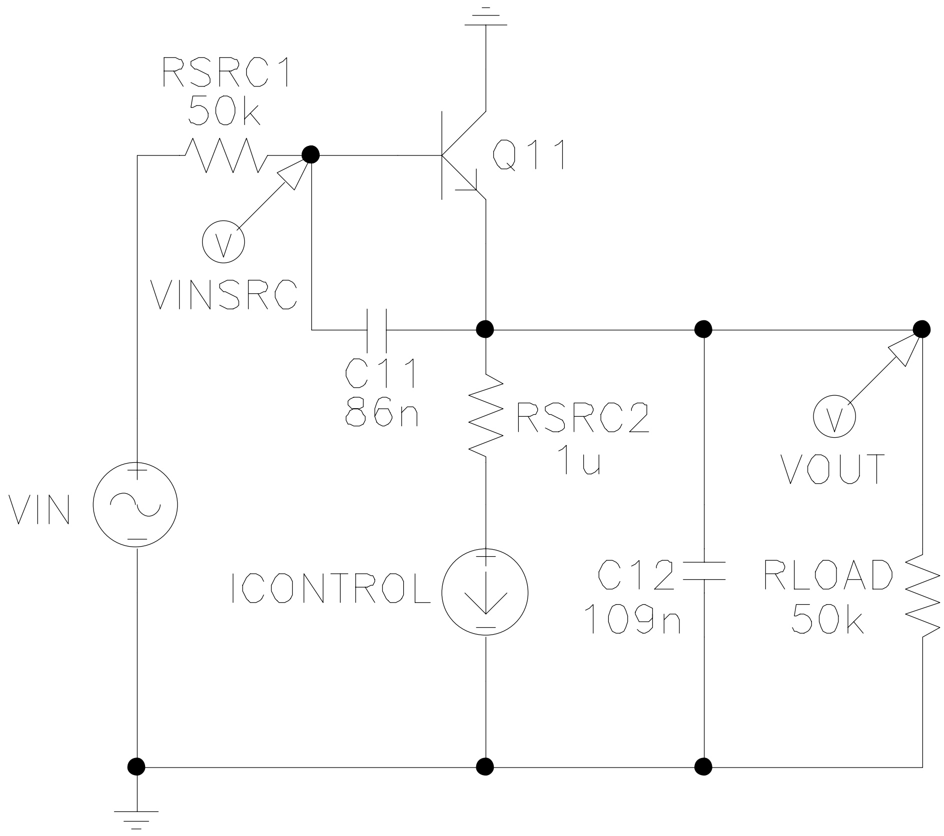
POST-2000 PATENTED INVENTIONS

LOW-VOLTAGE BALUN CIRCUIT
BEST EVOLVED FROM GENERATION 84



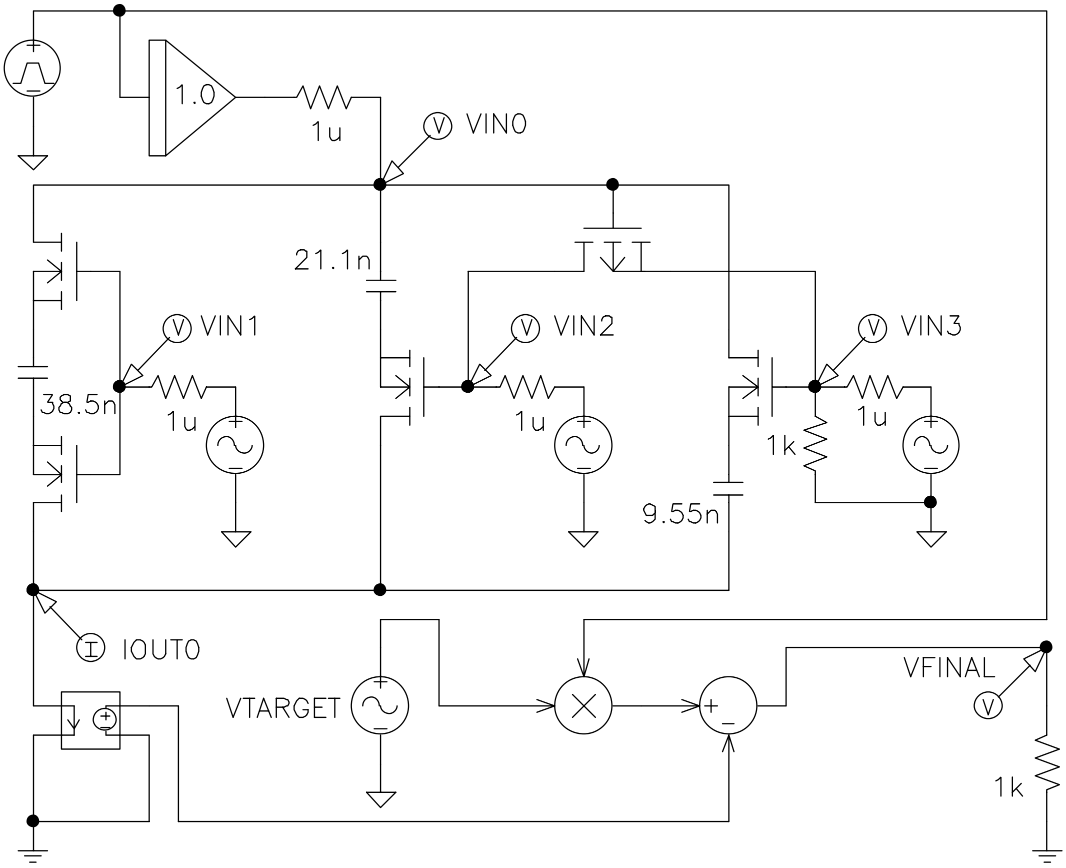
POST-2000 PATENTED INVENTIONS

TUNABLE INTEGRATED ACTIVE FILTER — GENERATION 50



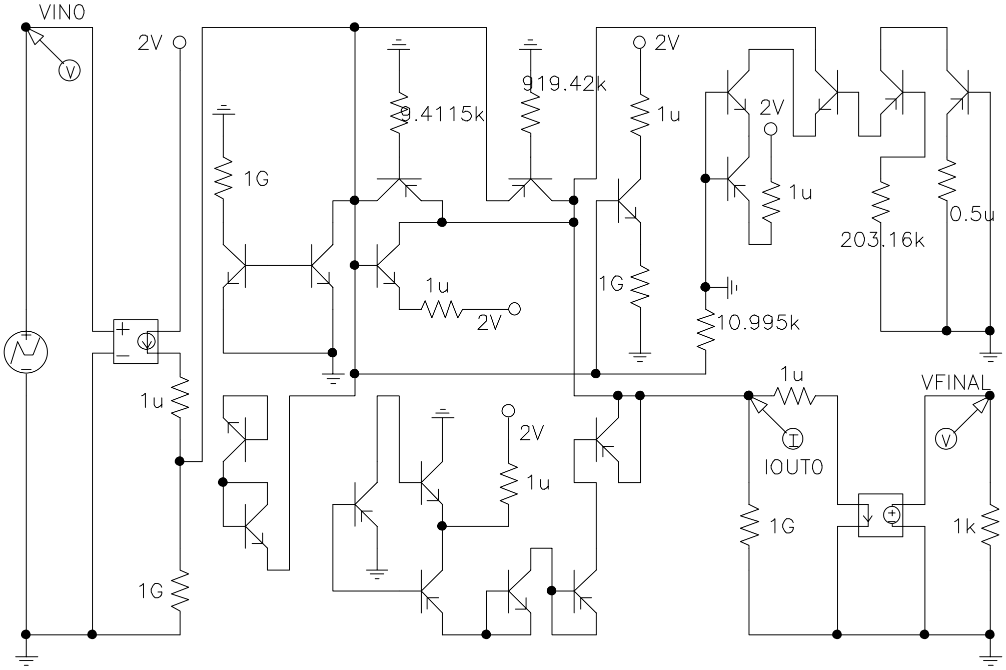
POST-2000 PATENTED INVENTIONS

REGISTER-CONTROLLED CAPACITOR CIRCUIT —GENERATION 98



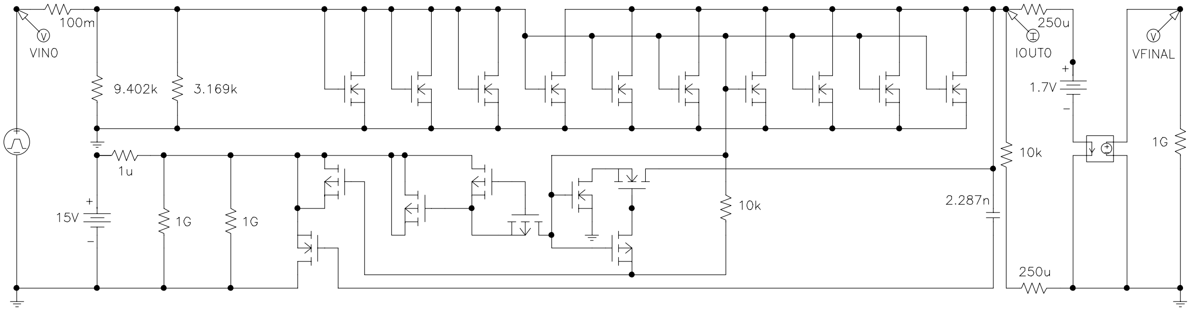
POST-2000 PATENTED INVENTIONS

LOW-VOLTAGE CUBIC SIGNAL GENERATION CIRCUIT FROM GENERATION 182



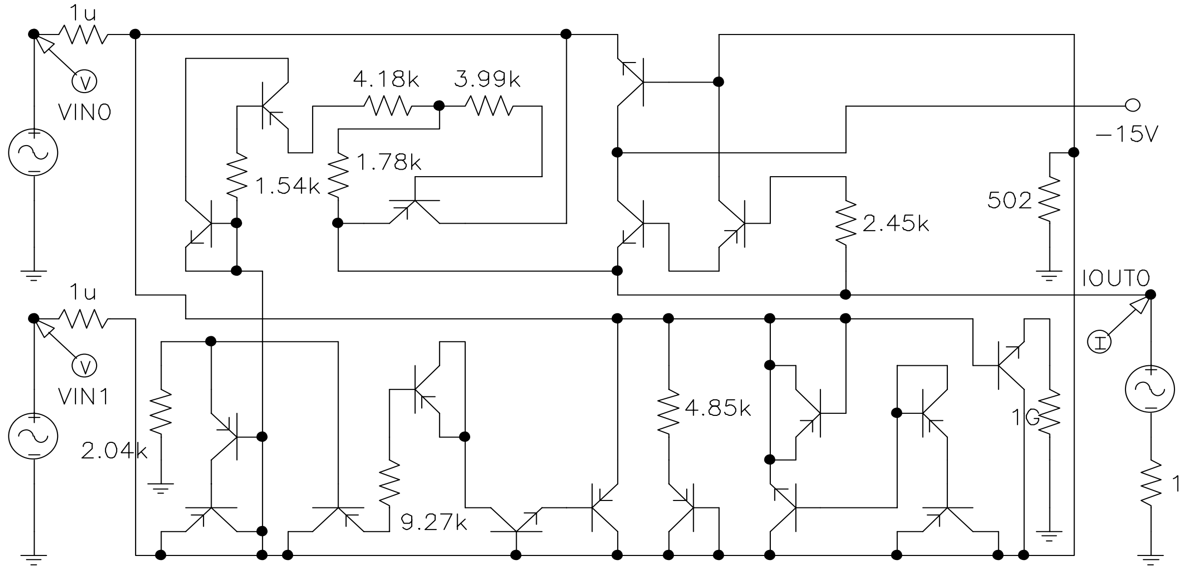
POST-2000 PATENTED INVENTIONS

HIGH CURRENT LOAD CIRCUIT FROM GENERATION 114



POST-2000 PATENTED INVENTIONS

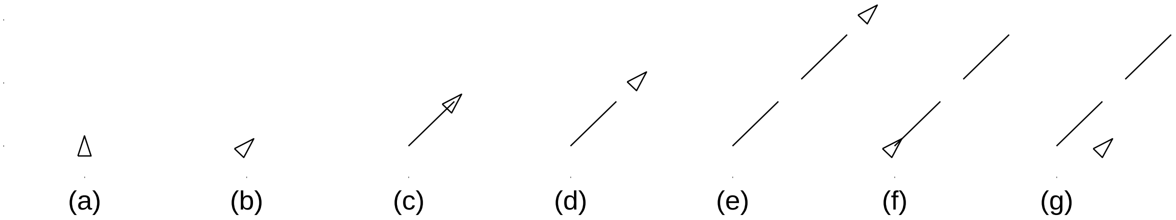
VOLTAGE-CURRENT-CONVERSION CIRCUIT FROM GENERATION 109



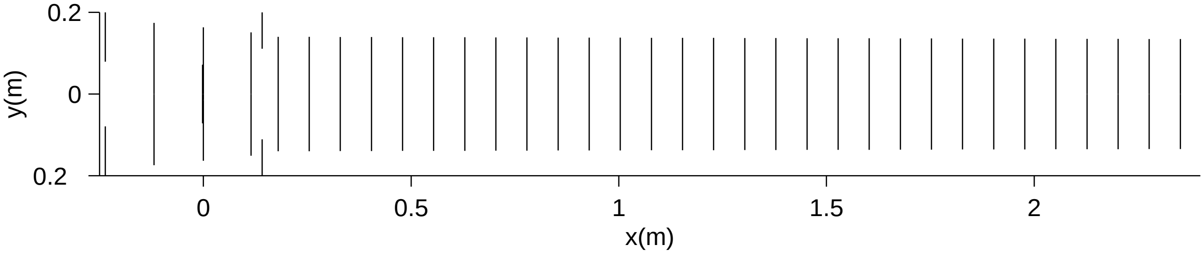
AUTOMATIC SYNTHESIS OF A WIRE ANTENNA

EXAMPLE OF TURTLE FUNCTIONS USED TO CREATE WIRE ANTENNA

```
(PROGN3
  (TURN-RIGHT 0.125)
  (LANDMARK
    (REPEAT 2
      (PROGN2
        (DRAW 1.0 HALF-MM-WIRE)
        (DRAW 0.5 NO-WIRE)))
  (TRANSLATE-RIGHT 0.125 0.75))
```



BEST-OF-RUN ANTENNA FROM GENERATION 90



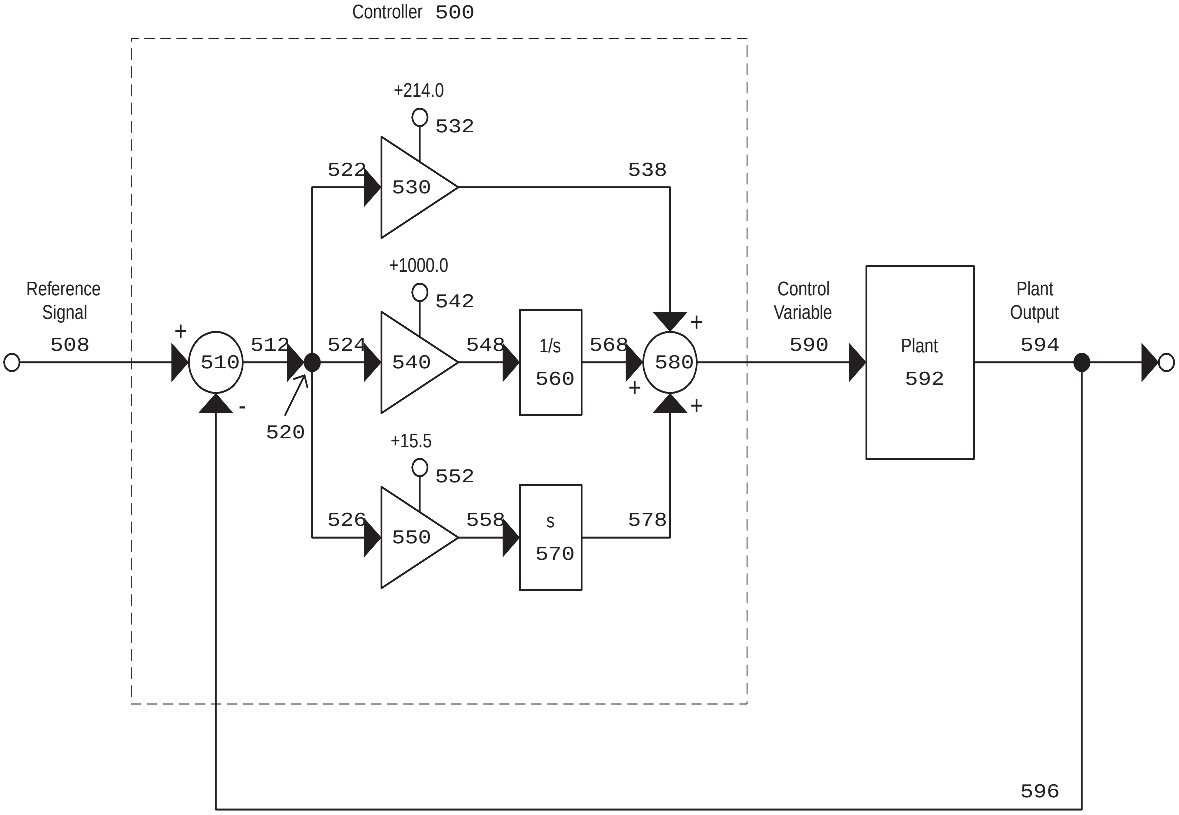
- **The GP run discovered**
  (1) the number of reflectors (one),
  (2) the number of directors,
  (3) the fact that the driven element, the directors, and the reflector are all straight wires,
  (4) the fact that the driven element, the directors, and the reflector are parallel,
  (5) the fact that the energy source (the transmission line) is connected only to single straight wire (the driven element) — that is, all the directors and reflectors are parasitically coupled

BEST-OF-RUN ANTENNA FROM GENERATION 90

- **Characteristics (3), (4), and (5) are essential characteristics of the Yagi-Uda antenna, namely an antenna with multiple parallel parasitically coupled straight-line directors, a single parallel parasitically coupled straight-line reflector, and a straight-line driven element.**

PID CONTROLLER

Block diagram of a plant and a PID controller composed of proportional (P), integrative (I), and derivative (D) blocks



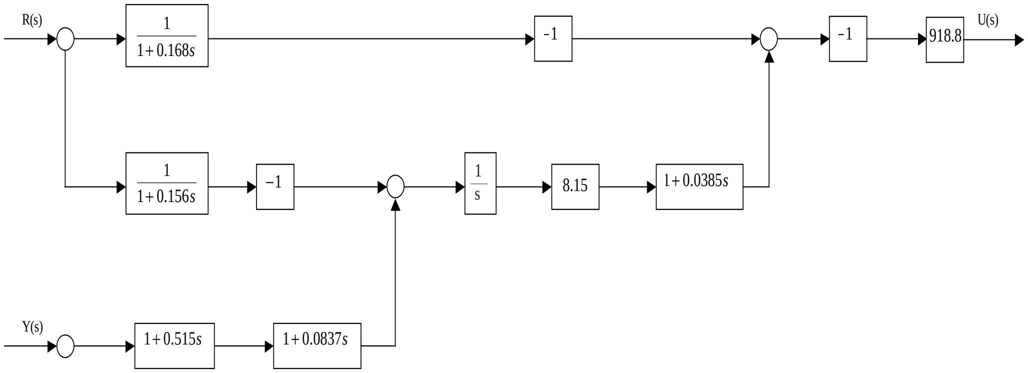
FUNCTION SET AND TERMINAL SET FOR TWO-LAG PLANT PROBLEM

FUNCTION SET AND TERMINAL SET

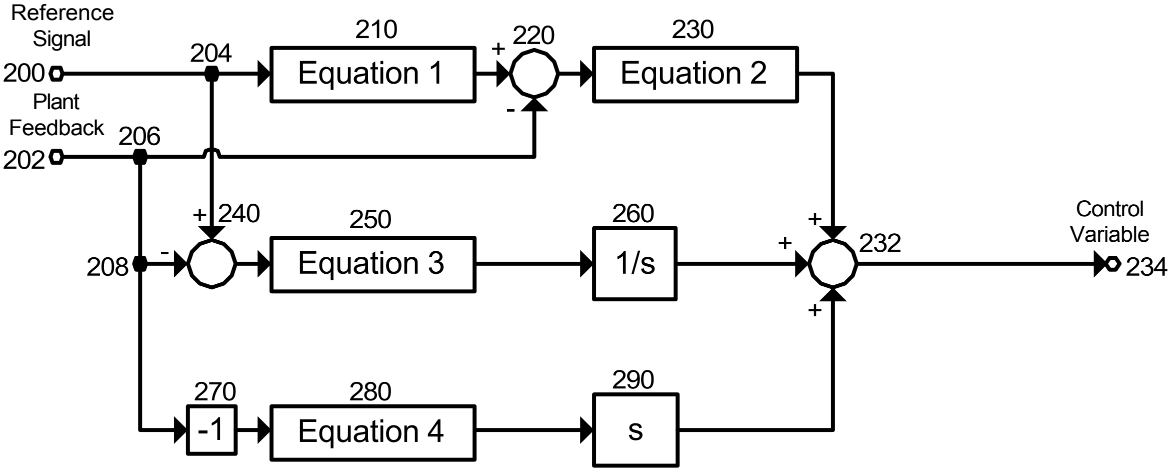
F = {**GAIN**, **INVERTER**, **LEAD**, **LAG**, **LAG2**, **DIFFERENTIAL_INPUT_INTEGRATOR**, **DIFFERENTIATOR**, **ADD_SIGNAL**, **SUB_SIGNAL**, **ADD_3_SIGNAL**, **ADF0**, **ADF1**, **ADF2**, **ADF3**, **ADF4**}

T = { **REFERENCE_SIGNAL**, **CONTROLLER_OUTPUT**, **PLANT_OUTPUT**, **CONSTANT_0**}

BEST-OF-RUN GENETICALLY EVOLVED CONTROLLER FROM GENERATION 32 FOR THE TWO-LAG PLANT



TOPOLOGY OF A PID CONTROLLER WITH NONZERO SETPOINT WEIGHTING OF THE REFERENCE SIGNAL IN THE PROPORTIONAL BLOCK BUT NO SETPOINT WEIGHTING FOR THE DERIVATIVE BLOCK



EVOLVED PID TUNING RULES

- **the proportional part of the controller**

$$K_{p\text{-}final} = 0.72 * K_u * e^{\frac{-1.6}{K_u} + \frac{1.2}{K_u^2}} - .0012340 * T_u - 6.1173 * 10^{-6}$$

- **the integrative part**

$$K_{i\text{-}final} = \frac{0.72 * K_u * e^{\frac{-1.6}{K_u} + \frac{1.2}{K_u^2}}}{0.59 * T_u * e^{\frac{-1.3}{K_u} + \frac{0.38}{K_u^2}}} - .068525 * \frac{K_u}{T_u}$$

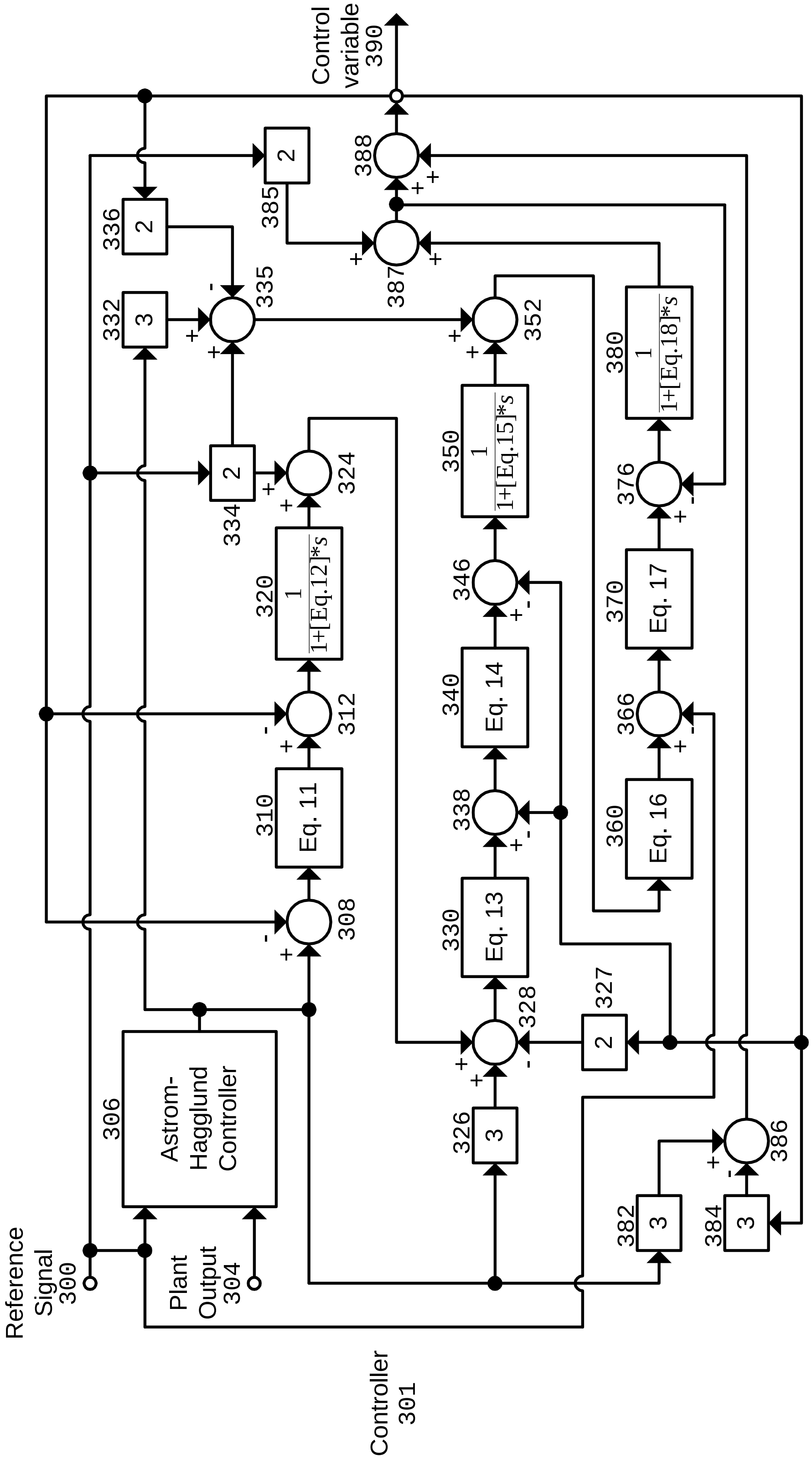
- **the derivative part**

$$K_{d\text{-}final} = 0.108 * K_u * T_u * e^{\frac{-1.6}{K_u} + \frac{1.2}{K_u^2}} * e^{\frac{-1.4}{K_u} + \frac{0.56}{K_u^2}} - 0.0026640 \left(e^{T_u}\right)^{\log\left(1.6342^{\log K_u}\right)}$$

- ***b_{final}*, setpoint weighting of the reference signal in the proportional block**

$$b_{final} = 0.25 * e^{\frac{0.56}{K_u} + \frac{-0.12}{K_u^2}} + \frac{K_u}{e^{K_u}}$$

IMPROVED NON PID CONTOLLERS



IMPROVED NON PID CONTOLLERS — CONTINUED

- **Gain block 310 is parameterized by equation 11:**

$$10^{e^{\log\left|\log\left|\log(e^{K_u*L})/L\right|\right|}} \quad [11]$$

- **Gain block 330 is parameterized by equation 13:**

$$10^{e^{\log\left|\log\left|K_u*L\right|\right|}} \quad [13]$$

- **Gain block 340 is parameterized by equation 14:**

$$e^{\log\left|K_u/L\right|} \quad [14]$$

- **Gain block 360 is parameterized by equation 16:**

$$10^{e^{\log\left|\log\left|K_u*L\right|\right|}} \quad [16]$$

- **Gain block 370 in figure 13.1 is parameterized by equation 17:**

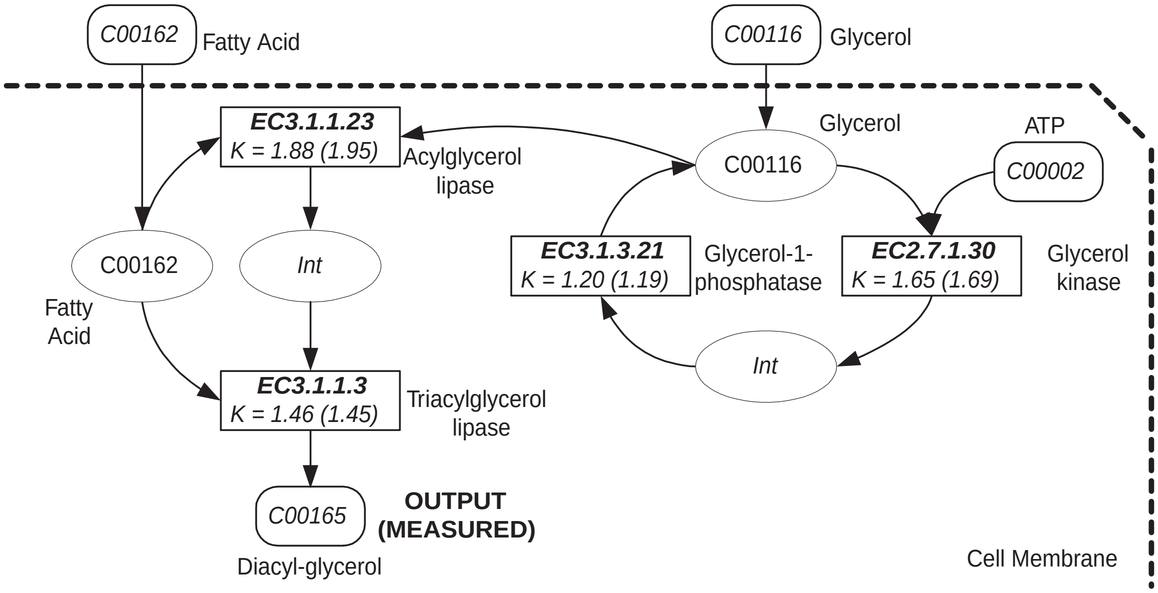
$$e^{\log(K_u)} \quad [17]$$

- **Equation 12 for lag block 320, equation 15 for lag block 350, and equation 18 for lag block 380 are the same:**

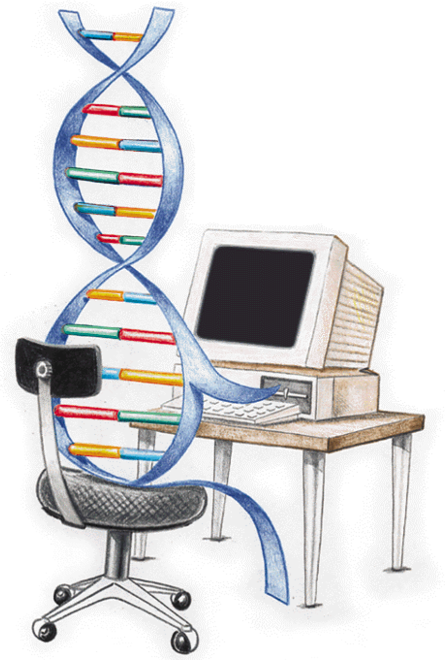
$$T_r$$

REVERSE ENGINEERING OF METABOLIC PATHWAYS

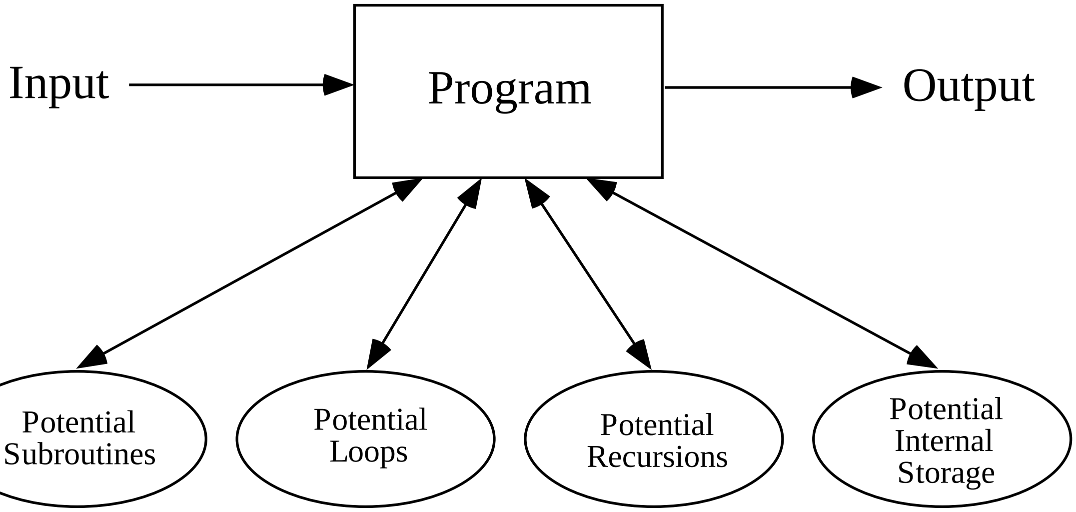
BEST-OF-GENERATION 66



REUSE



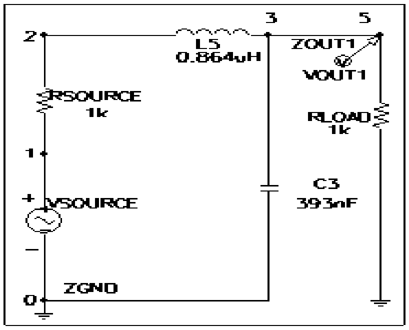
A COMPUTER PROGRAM



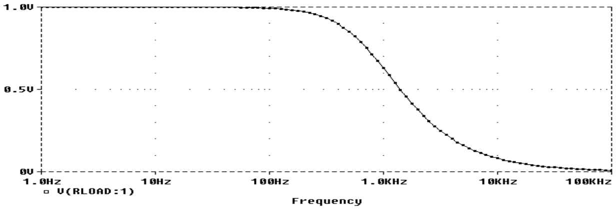
- **Subroutines provide one way to REUSE code — possibly with different instantiations of the dummy variables (formal parameters)**
- **Loops (and iterations) provide a 2nd way to REUSE code**
- **Recursion provide a 3rd way to REUSE code**
- **Memory provides a 4th way — to REUSE the results of executing code**

REUSE
LOWPASS FILTER USING ADFs

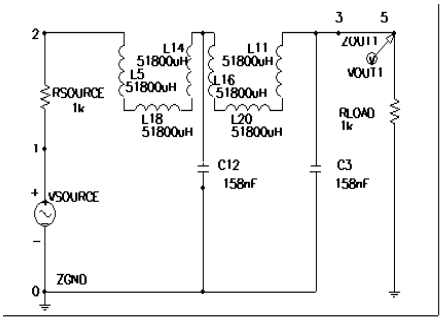
GENERATION 0 – ONE-RUNG LADDER



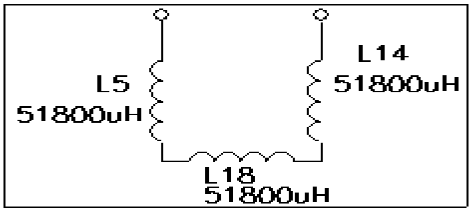
BEHAVIOR IN FREQUENCY DOMAIN



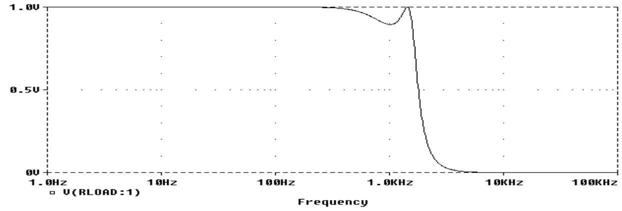
REUSE
LOWPASS FILTER USING ADFs
GENERATION 9 - TWO-RUNG LADDER



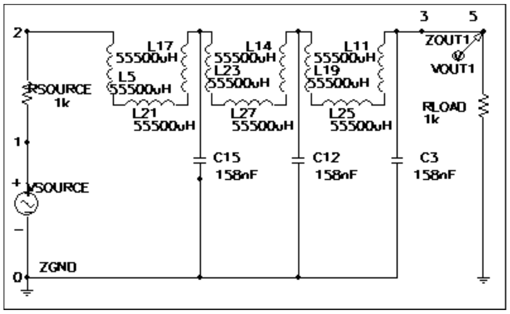
TWICE-CALLED TWO-PORTED `ADF0`



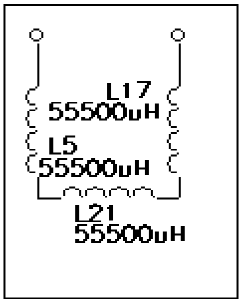
BEHAVIOR IN FREQUENCY DOMAIN



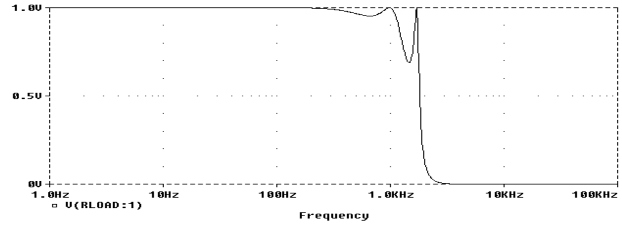
REUSE
LOWPASS FILTER USING ADFs
GEN 16 – THREE-RUNG LADDER



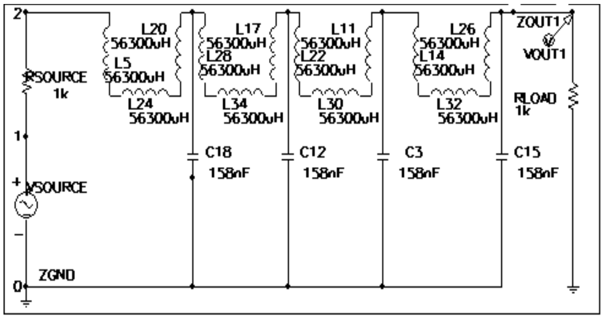
THRICE-CALLED TWO-PORTED `ADF0`



BEHAVIOR IN FREQUENCY DOMAIN

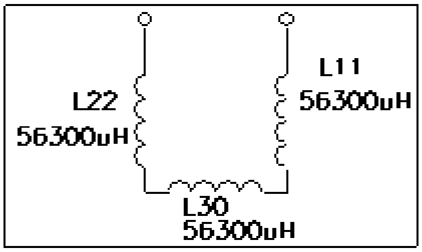


REUSE
LOWPASS FILTER USING ADFS
GEN 20 – FOUR-RUNG LADDER

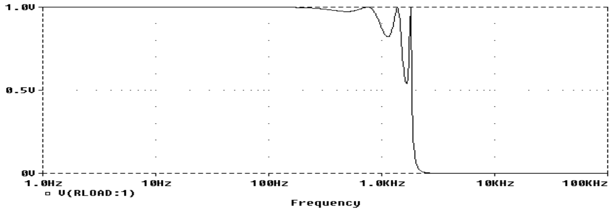


QUADRUPLY-CALLED TWO-PORTED

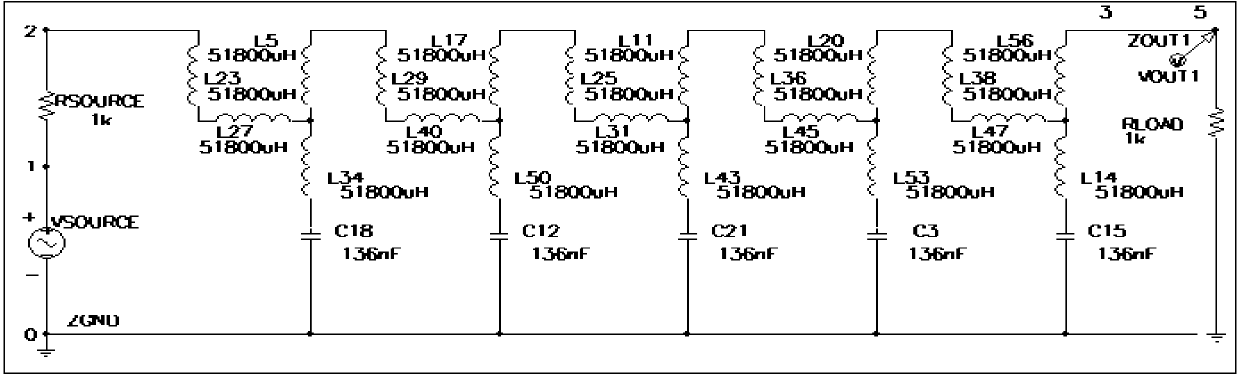
`ADF0`



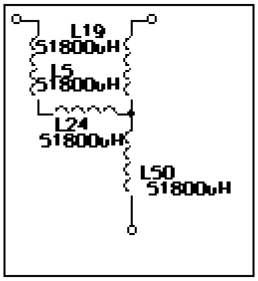
BEHAVIOR IN FREQUENCY DOMAIN



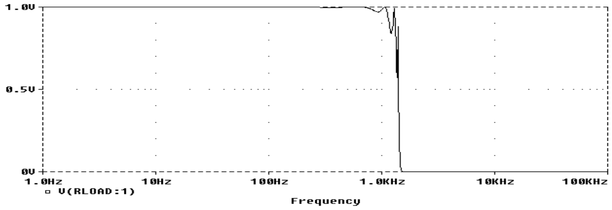
REUSE
LOWPASS FILTER USING ADFs
GENERATION 31 — TOPOLOGY OF CAUER (ELLIPTIC) FILTER



QUINTUPLY-CALLED THREE-PORTED ADF0



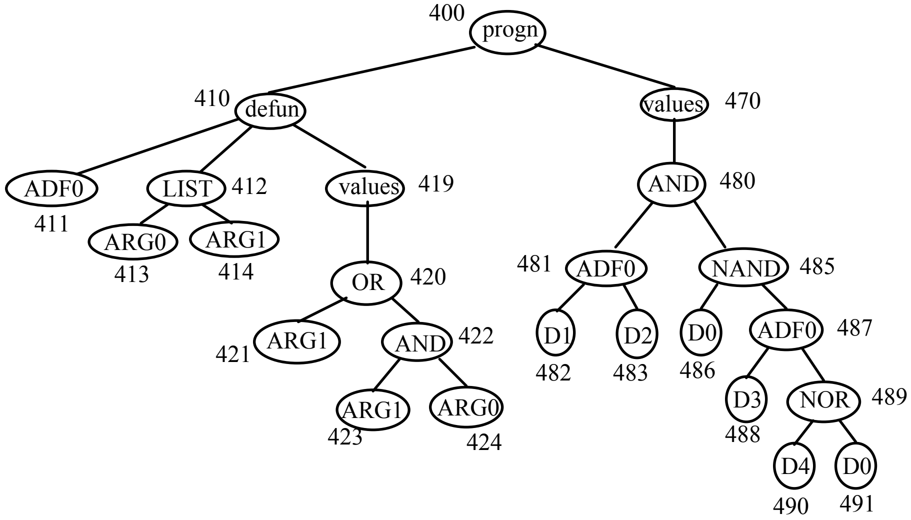
BEHAVIOR IN FREQUENCY DOMAIN



ARCHITECTURE

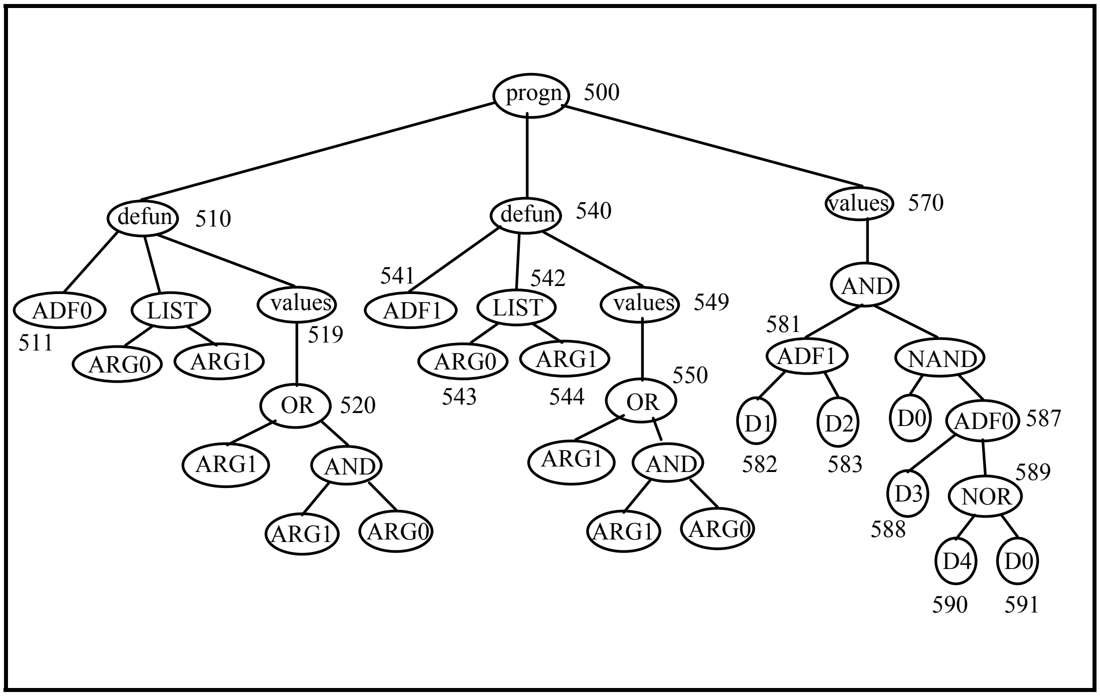
ARCHITECTURE-ALTERING OPERATIONS

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



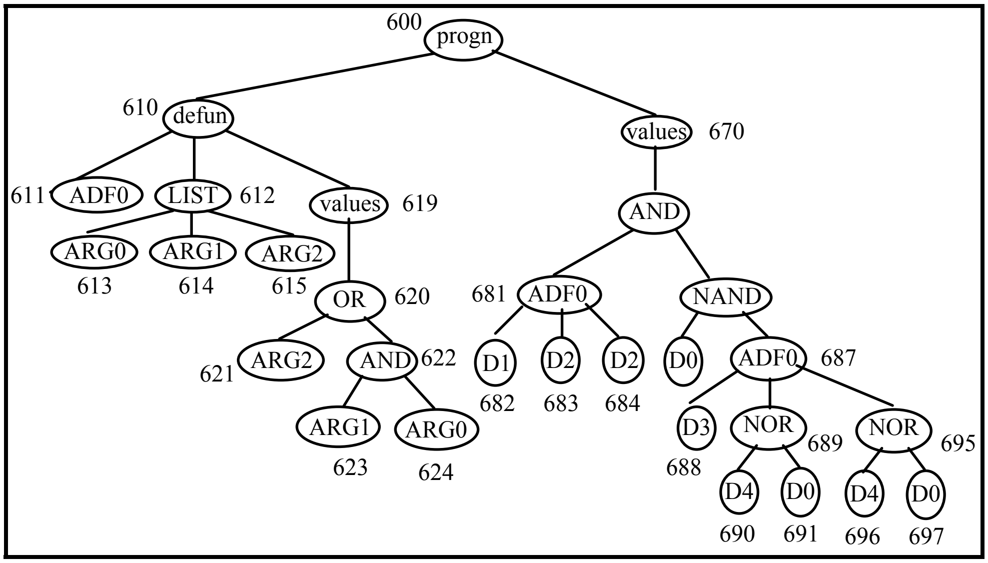
ARCHITECTURE-ALTERING OPERATIONS

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

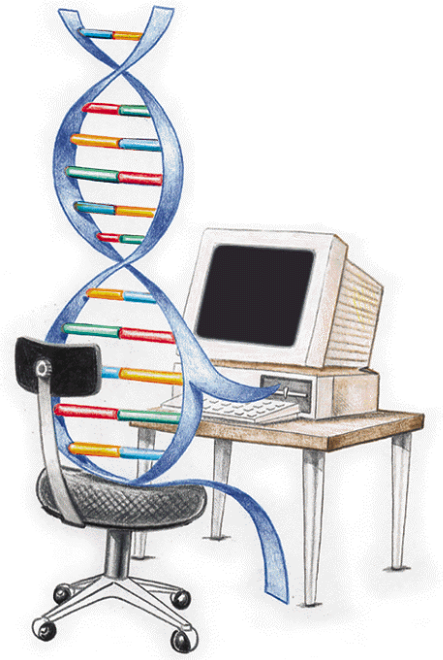


ARCHITECTURE-ALTERING OPERATIONS

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



PARAMETER PASSING



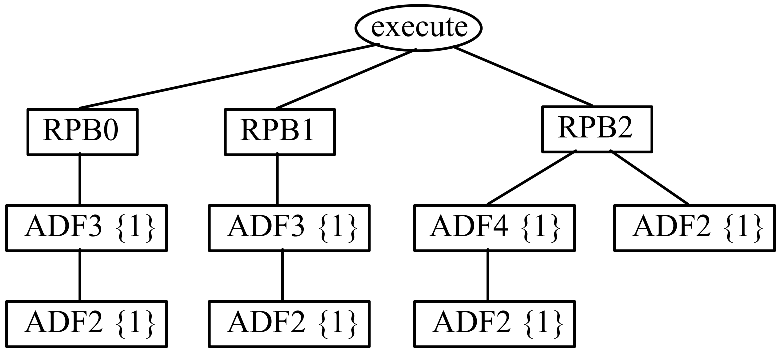
PASSING A PARAMETER TO A SUBSTRUCTURE

- **The set of potential terminals for each construction-continuing subtree of an automatically defined function, $T_{ccs\text{-}adf\text{-}potential}$, is**

$T_{ccs\text{-}adf\text{-}potential}$ = {`ARG0`}

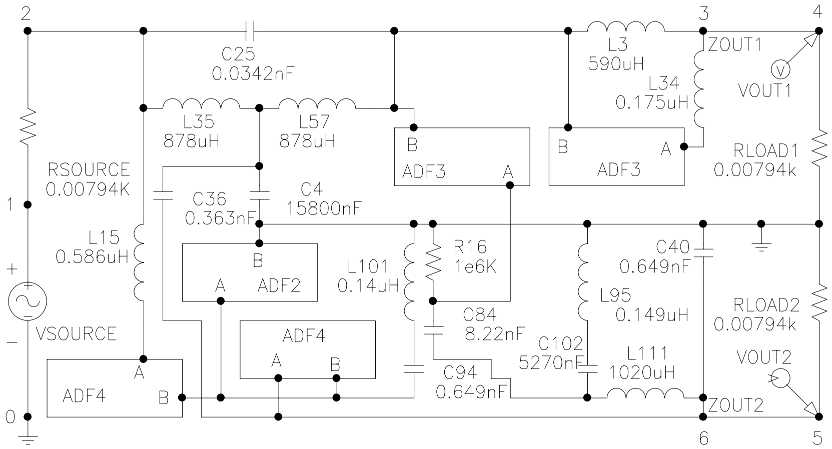
EMERGENCE OF A PARAMETERIZED ARGUMENT IN A CIRCUIT SUBSTRUCTURE

HIERARCHY OF BRANCHES FOR THE BEST-OF-RUN CIRCUIT- FROM GENERATION 158



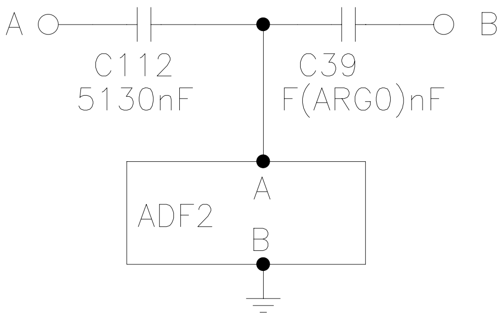
PASSING A PARAMETER TO A SUBSTRUCTURE

BEST-OF-RUN CIRCUIT FROM GENERATION 158



THREE-PORTED AUTOMATICALLY DEFINED FUNCTION **ADF3** OF THE BEST-OF-RUN CIRCUIT FROM GENERATION 158

ADF3 CONTAINS CAPACITOR **C39** PARAMETERIZED BY DUMMY VARIABLE **ARG0**



THE FIRST RESULT-PRODUCING BRANCH, RPB0, CALLING ADF3

```
(PARALLEL0 (L (+ (– 1.883196E-01 (– -9.095883E-02 5.724576E-
01)) (– 9.737455E-01 -9.452780E-01)) (FLIP END)) (SERIES (C (+
(+ -6.668774E-01 -8.770285E-01) 4.587758E-02) (NOP END))
(SERIES END END (PARALLEL1 END END END END)) (FLIP
(SAFE_CUT))) (PAIR_CONNECT_0 END END END) (PAIR_CONNECT_0 (L
(+ -7.220122E-01 4.896697E-01) END) (L (– -7.195599E-01
3.651142E-02) (SERIES (C (+ -5.111248E-01 (– (– -6.137950E-01
-5.111248E-01) (– 1.883196E-01 (– -9.095883E-02 5.724576E-
01)))) END) (SERIES END END (adf3 6.196514E-01)) (NOP END)))
(NOP END)))
```

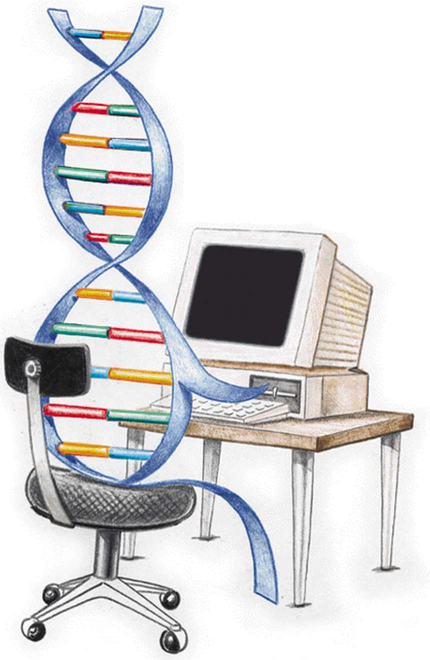
AUTOMATICALLY DEFINED FUNCTION ADF3

```
(C (+ (– (+ (+ (+ 5.630820E-01 (– 9.737455E-01 -9.452780E-01))
(+ ARG0 6.953752E-02)) (– (– 5.627716E-02 (+ 2.273517E-01 (+
1.883196E-01 (+ 9.346950E-02 (+ -7.220122E-01 (+ 2.710414E-02
1.397491E-02)))))) (– (+ (– 2.710414E-02 -2.807583E-01) (+ -
6.137950E-01 -8.554120E-01)) (– -8.770285E-01 (– -4.049602E-01
-2.192044E-02))))) (+ (+ 1.883196E-01 (+ (+ (+ (+ 9.346950E-02
(+ -7.220122E-01 (+ 2.710414E-02 1.397491E-02))) (– 4.587758E-
02 -2.340137E-01)) 3.226026E-01) (+ -7.220122E-01 (– -
9.131658E-01 6.595502E-01)))) 3.660116E-01)) 9.496355E-01)
(THREE_GROUND_0 (C (+ (– (+ (+ (+ 5.630820E-01 (– 9.737455E-01
-9.452780E-01)) (+ (– (– -7.195599E-01 3.651142E-02) -
9.761651E-01) (– (+ (– (– -7.195599E-01 3.651142E-02) -
9.761651E-01) 6.953752E-02) 3.651142E-02))) (– (– 5.627716E-02
(– 1.883196E-01 (– -9.095883E-02 5.724576E-01))) (– (+ (–
2.710414E-02 -2.807583E-01) (+ -6.137950E-01 (+ ARG0
6.953752E-02))) (– -8.770285E-01 (– -4.049602E-01 -2.192044E-
02))))) (+ (+ 1.883196E-01 -7.195599E-01) 3.660116E-01))
9.496355E-01) (NOP (FLIP (PAIR_CONNECT_0 END END END)))) (FLIP
(SERIES (FLIP (FLIP (FLIP END))) (C (– (+ 6.238477E-01
6.196514E-01) (+ (+ (– (– 4.037348E-01 4.343444E-01) (+ -
7.788187E-01 (+ (+ (– -8.786904E-01 1.397491E-02) (– -
6.137950E-01 (– (+ (– 2.710414E-02 -2.807583E-01) (+ -
6.137950E-01 -8.554120E-01)) (– -8.770285E-01 (– -4.049602E-01
-2.192044E-02))))) (+ (+ 7.215142E-03 1.883196E-01) (+
7.733750E-01 4.343444E-01))))) (– (– -9.389297E-01 5.630820E-
01) (+ -5.840433E-02 3.568947E-01))) -8.554120E-01)) (NOP
END)) END)) (FLIP (adf2 9.737455E-01))))
```

ADF3 DOES THREE THINGS

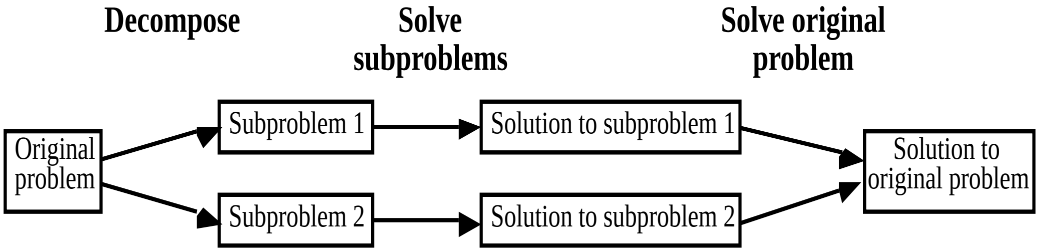
- The structure that develops out of **ADF3** includes a capacitor **C112** whose value (5,130 uF) is not a function of its dummy variable, **ARG0**.
- The structure that develops out of **ADF3** has one hierarchical reference to **ADF2**. The invocation of **ADF2** is done with a constant (9.737455E-01) and produces a 259 μH inductor.
- Most importantly, the structure that develops out of **ADF3** creates a capacitor (**C39**) whose sizing, **F(ARG0)**, is a function of the dummy variable, **ARG0**, of automatically defined function **ADF3**. Capacitor **C39** has different sizing on different invocations of automatically defined function **ADF3**.

PROBLEM DECOMPOSITION



TOP-DOWN VIEW OF THREE STEP HIERARCHICAL PROBLEM-SOLVING PROCESS

DIVIDE AND CONQUER

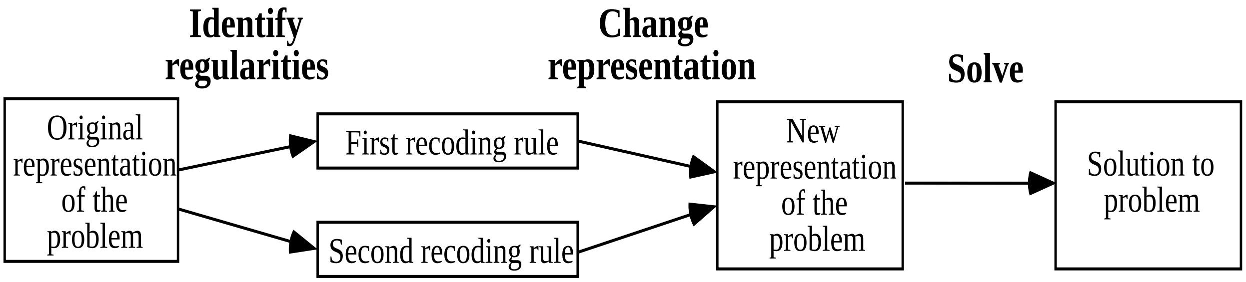


- **Decompose a problem into subproblems**

- **Solve the subproblems**

- **Assemble the solutions of the subproblems into a solution for the overall problem**

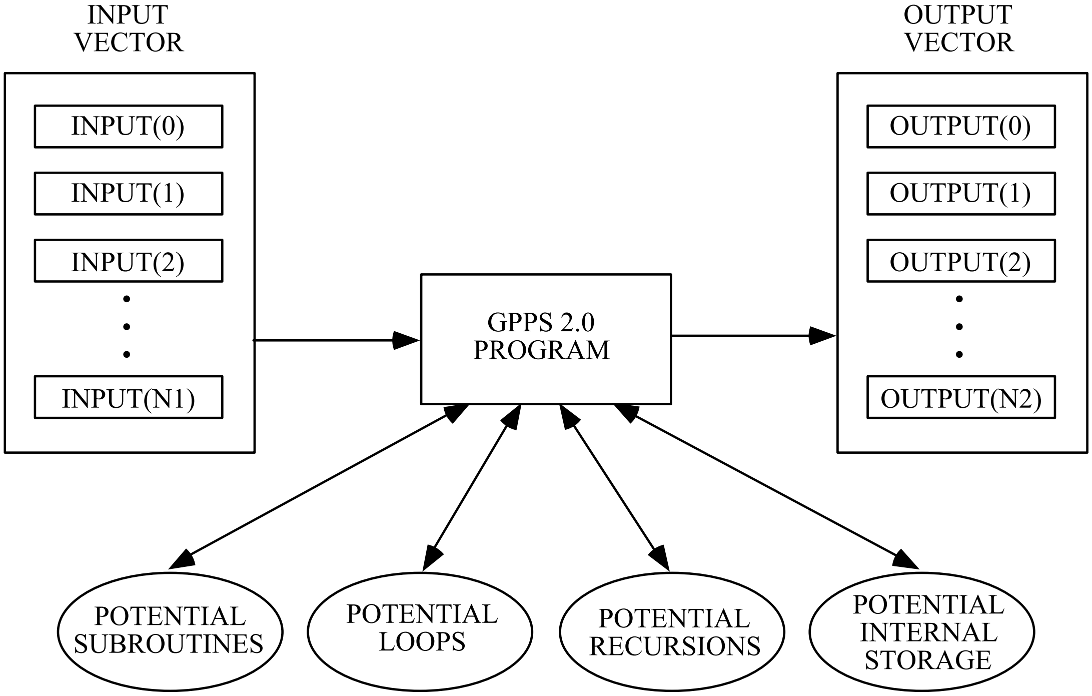
BOTTOM-UP VIEW OF THREE STEP HIERARCHICAL PROBLEM-SOLVING PROCESS



- **Identify regularities**
- **Change the representation**
- **Solve the overall problem**

AUTOMATIC PROGRAMMING

GENETIC PROGRAMMING PROBLEM SOLVER (GPPS) — VERSION 2.0



16 ATTRIBUTES OF A SYSTEM FOR AUTOMATICALLY CREATING COMPUTER PROGRAMS

1 — Starts with "What needs to be done"
2 — Tells us "How to do it"
3 — Produces a computer program
4 — Automatic determination of program size
5 — Code reuse
6 — Parameterized reuse
7 — Internal storage
8 — Iterations, loops, and recursions
9 — Self-organization of hierarchies
10 — Automatic determination of program architecture
11 — Wide range of programming constructs
12 — Well-defined
13 — Problem-independent
14 — Wide applicability
15 — Scalable
16 — Competitive with human-produced results

GENETIC PROGRAMMING OVER 15-YEAR PERIOD 1987–2002

System	Period of usage	Petacycles (10^{15}cycles) per day for entire system	Speed-up over previous system	Speed-up over first system in this table	Human-competitive results
Serial Texas Instruments LISP machine	**1987–1994**	**0.00216**	**1 (base)**	**1 (base)**	**0**
64-node Transtech transputer parallel machine	**1994–1997**	**0.02**	**9**	**9**	**2**
64-node Parsytec parallel machine	**1995–2000**	**0.44**	**22**	**204**	**12**
70-node Alpha parallel machine	**1999–2001**	**3.2**	**7.3**	**1,481**	**2**
1,000-node Pentium II parallel machine	**2000–2002**	**30.0**	**9.4**	**13,900**	**12**

PROGRESSION OF RESULTS

System	Period	Speed-up	Qualitative nature of the results produced by genetic programming
Serial LISP machine	1987–1994	1 (base)	• Toy problems of the 1980s and early 1990s from the fields of artificial intelligence and machine learning
64-node Transtech 8-biy transputer	1994–1997	9	•Two human-competitive results involving one-dimensional discrete data (not patent-related)
64-node Parsytec parallel machine	1995–2000	22	• One human-competitive result involving two-dimensional discrete data • Numerous human-competitive results involving continuous signals analyzed in the frequency domain • Numerous human-competitive results involving 20th-century patented inventions
70-node Alpha parallel machine	1999–2001	7.3	• One human-competitive result involving continuous signals analyzed in the time domain • Circuit synthesis extended from topology and sizing to include routing and placement (layout)
1,000-node Pentium II parallel machine	2000–2002	9.4	• Numerous human-competitive results involving continuous signals analyzed in the time domain • Numerous general solutions to problems in the form of parameterized topologies • Six human-competitive results duplicating the functionality of 21st-century patented inventions
Long (4-week) runs of 1,000-node Pentium II parallel machine	2002	9.3	• Generation of two patentable new inventions

PROGRESSION OF QUALITATIVELY MORE SUBSTANTIAL RESULTS PRODUCED BY GENETIC PROGRAMMING IN RELATION TO FIVE ORDER-OF-MAGNITUDE INCREASES IN COMPUTATIONAL POWER

- **toy problems**
- **human-competitive results not related to patented inventions**
- **20th-century patented inventions**
- **21st-century patented inventions**
- **patentable new inventions**

THE CHALLENGE OF AUTOMATIC PROGRAMMING

"How can computers learn to solve problems without being explicitly programmed? In other words, how can computers be made to do what is needed to be done, without being told exactly how to do it?"

— Attributed to Arthur Samuel (1959)

CRITERION FOR SUCCESS

"The aim [is] ... to get machines to exhibit behavior, which if done by humans, would be assumed to involve the use of intelligence."

— Arthur Samuel (1983)