A Comparative Study of Two Boolean Formulations of FPGA Detailed Routing Constraints

Gi-Joon Nam, Fadi Aloul, Karem Sakallah and ‡Rob Rutenbar

Department of EECS
University of Michigan
{criteria, faloul, karem}@eecs.umich.edu

‡Department of ECE
Carnegie Mellon University
rutenbar@ece.cmu.edu

Outline

- Motivation
- About Boolean Satisfiability (SAT)
- Target FPGA Architecture
- Boolean SAT-based FPGA Detailed Routing Formulation
  - Track-based Formulation Method
  - Route-based Formulation Method
- Comparative Experiment Results
- Conclusions & Future Work

Motivation

- FPGA market is growing fast !
  - Lower prototype cost and shorter time-to-market
  - FPGA technology advances rapidly in density and speed
- Growing interest in Reconfigurable Computing
  - Require to reduce reconfiguration overhead
- Significant progress in SAT (satisfiability) realm !
  - Broad range of SAT applications in EDA area
  - SAT provides “exact” solutions
  - Various efficient SAT engines: GRASP, RELSAT, SATO etc.

About Boolean Satisfiability (SAT)

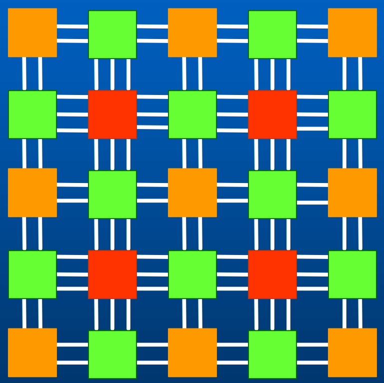
- Given a suitable representation for a Boolean function **f(X):**
  - Find an assignment **X*** such that **f(X*) = 1**
  - Or prove that such an assignment does not exist (i.e. f(X) = 0 for all possible assignments)
- In the “classical” SAT problem
  - **f(X)** is represented in **conjunctive normal form (CNF)** or product-of-sums (POS)

$$\varphi = (a \vee c) \wedge (b \vee c) \wedge (\neg a \vee \neg b \vee \neg c)$$

literal ***clause***

Routing Architecture Model

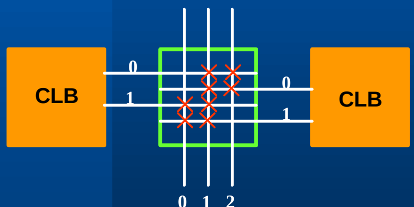
- Island-style FPGA fabric
  - Configurable logic block (CLB)
  - Connection block (C-block)
  - Switching block (S-block)



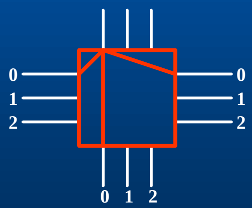
Island-Style FPGA

Routing Architecture Model

- Architectural parameters:
  - *W* : number of tracks / channel
  - *Fc* : flexibility of C-block
  - *Fs* : flexibility of S-block



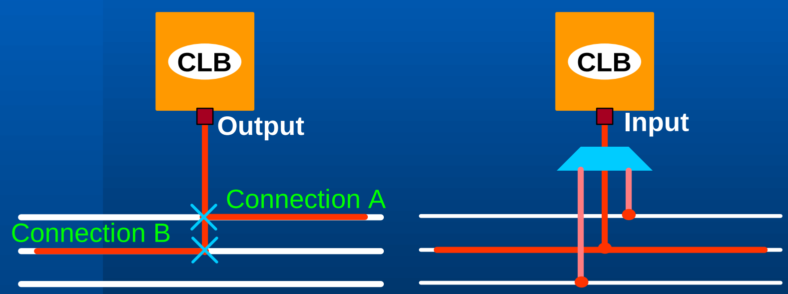
C-Block Fc = 2



S-Block Fs = 3

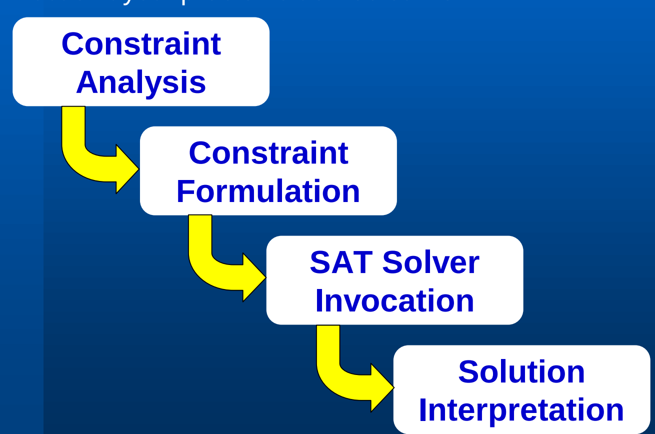
Pin Connection Issue

- Why two-pin connection based formulation…….

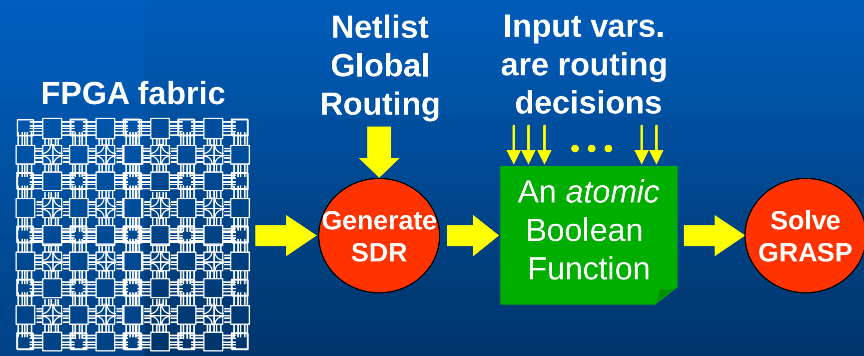


Boolean SAT-based Layout Approach

- Satisfiability-based layout formulation
  - Recast layout problems as Boolean SAT



Boolean SAT Formulation of FPGA Routing



Boolean SAT-based Layout Approach

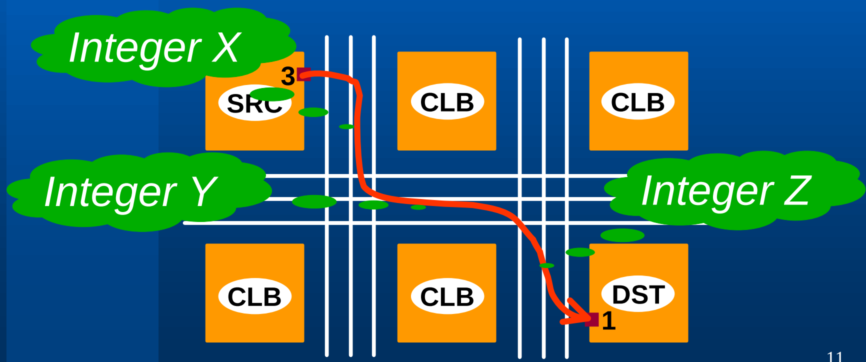
- Naturally decision (Yes/No) problem

Satisfiable = Routable
Unsatisfiable = Unroutable

- Simultaneous net embedding
- Exact method
  - Routability decision or Routability estimator
- Similar modelling ability as ILP (Integer Linear Programming)
- Proven to be more efficient than ILP in our application

Track-based Routing Formulation

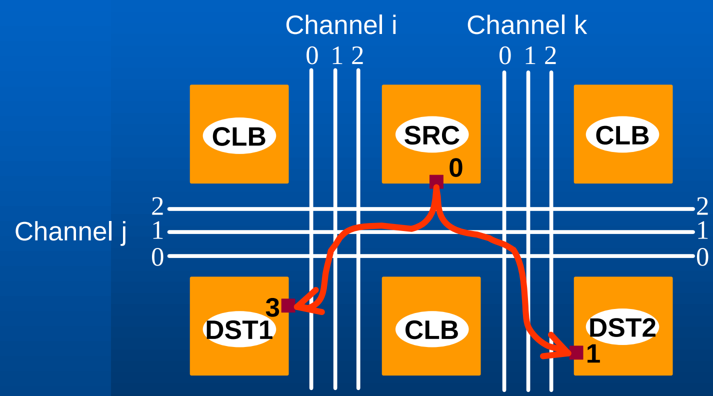
- Net segment-to-track assignment problem
- Each net is represented by a set of integer “track” variables



Track-based Routing Formulation

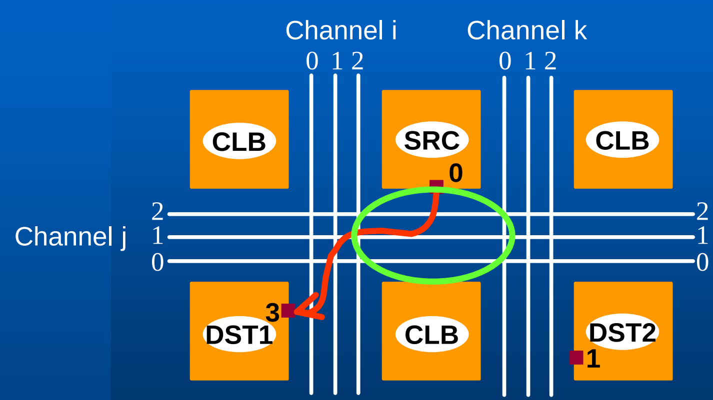
- Routing constraints
  - Connectivity constraint C
    - Each net connects through a set of legal, contiguous routing resources
  - Exclusivity constraint E
    - No two distinct nets share routing resources in any routing areas
- Boolean routing function: **R = C Ù E**

Connectivity Constraint



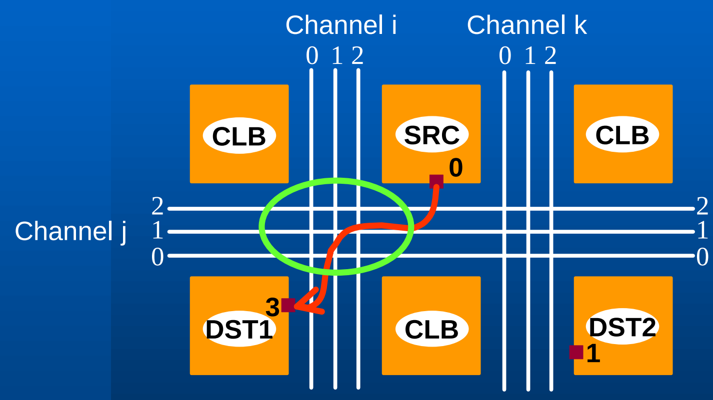
W = 3, Fc = 3, Fs = 3

Connectivity Constraint



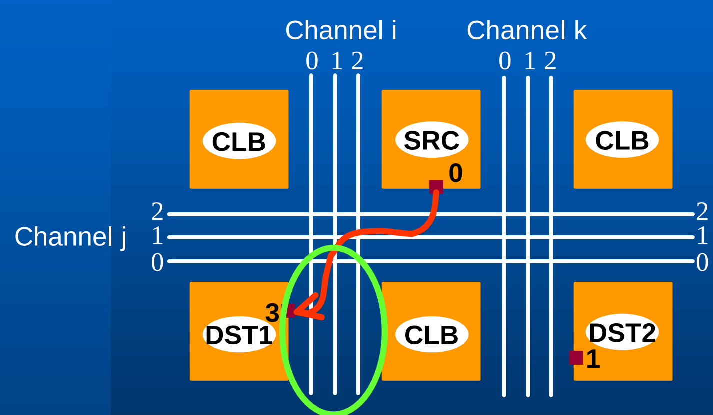
Conn1 = [(A1 = 0) Ú (A1 = 1) Ú (A1 = 2)]

Connectivity Constraint



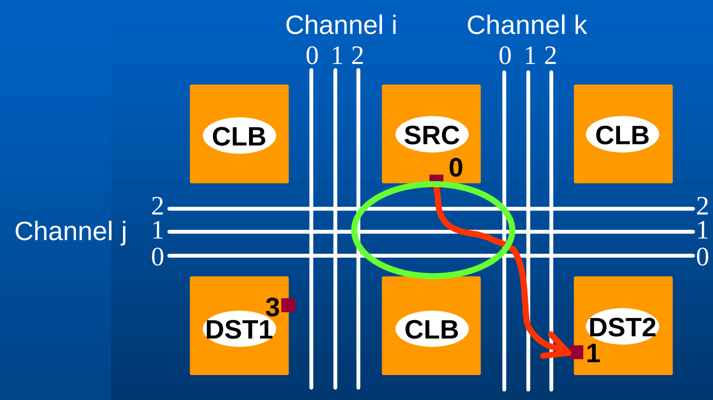
Conn2 = [(A1 = A2)]

Connectivity Constraint



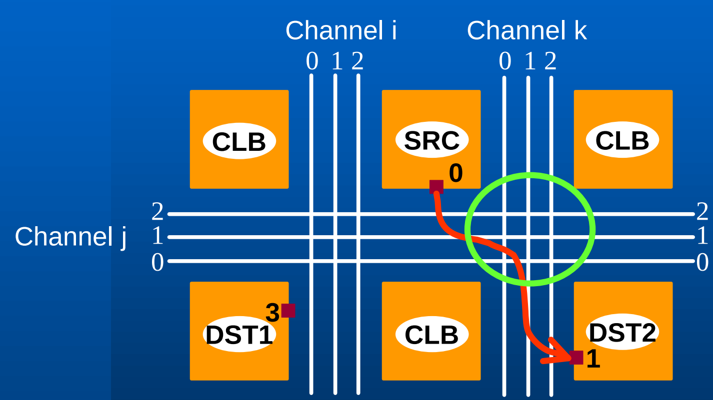
Conn3 = [(A2 = 0) Ú (A2 = 1) Ú (A2 = 2)]

Connectivity Constraint



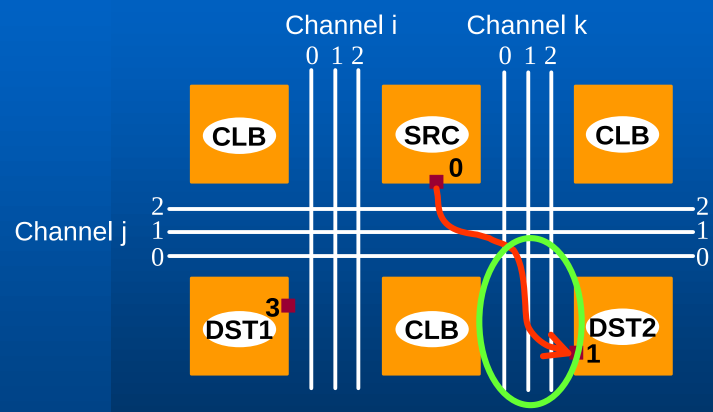
Conn4 = [(B1 = 0) Ú (B1 = 1) Ú (B1 = 2)]

Connectivity Constraint



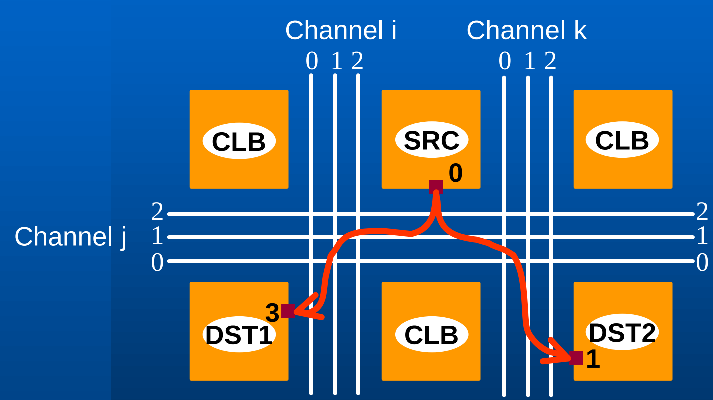
Conn5 = [(B1 = B2)]

Connectivity Constraint



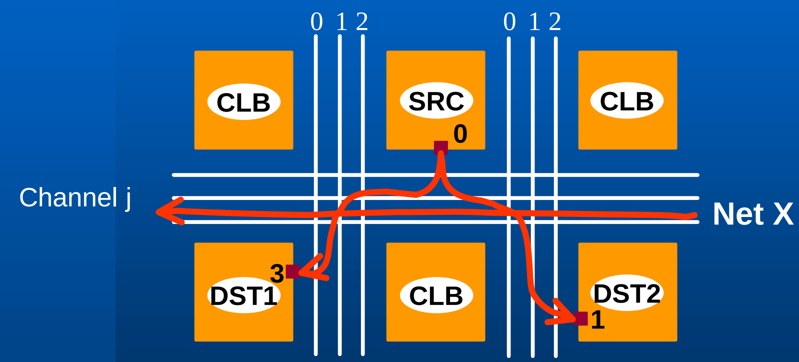
Conn6 = [(B2 = 0) Ú (B2 = 1) Ú (B2 = 2)]

Connectivity Constraint



Conn(N) = [Conn1 Ù Conn2 Ù Conn3 Ù Conn4 Ù Conn5 Ù Conn6]

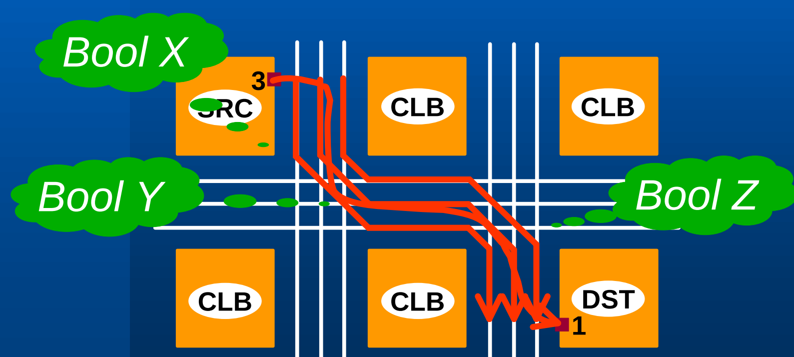
Exclusivity Constraint



Excl_m = [(A ¹ X) Ù (B ¹ X)]

Route-based Routing Formulation

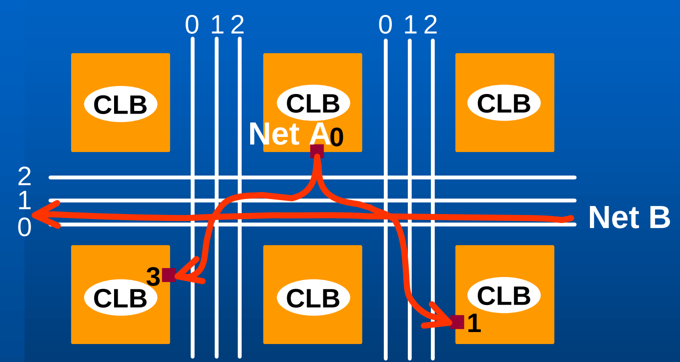
- Routability checking problem
- Each detailed route of a net is represented by a Boolean “selection” variable



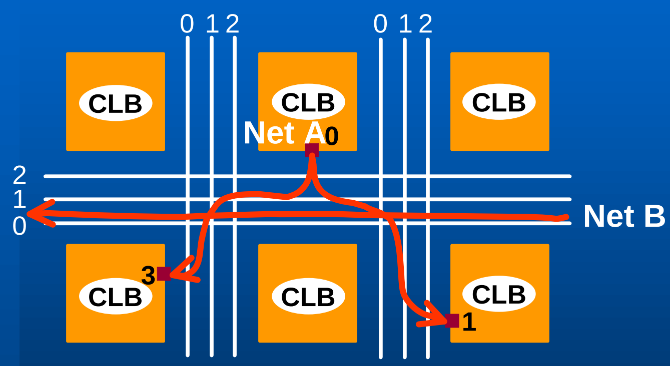
Route-based Routing Formulation

- Routing constraints
  - Liveness constraint L
    - Each two-pin connection has at least one detailed route selected in the routing solution
  - Exclusivity constraint E
    - No two distinct nets share routing resources in any routing areas
- Boolean routing function: **R = L Ù E**

Route-based Constraint Example



Liveness Constraint

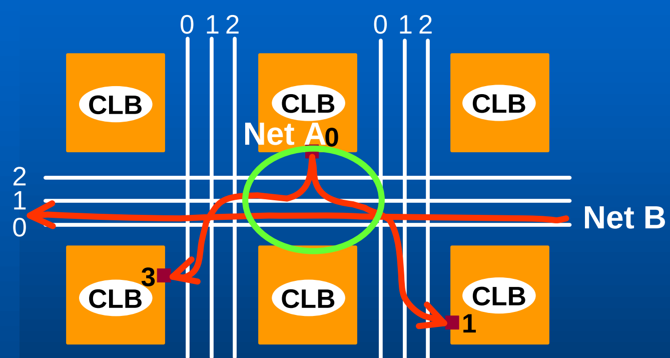


Live(A1) = [A1R0 Ú A1R1 Ú A1R2]

Live(A2) = [A2R0 Ú A2R1 Ú A2R2]

Live(B1) = [B1R0 Ú B1R1 Ú B1R2]

Exclusivity Constraint



Excl(Track0) = [(∅ A1R0 Ú ∅ B1R0) Ù (∅ A2R0 Ú ∅ B1R0)]

Excl(Track1) = [(∅ A1R1 Ú ∅ B1R1) Ù (∅ A2R1 Ú ∅ B1R1)]

Excl(Track2) = [(∅ A1R2 Ú ∅ B1R2) Ù (∅ A2R2 Ú ∅ B1R2)]

Why route-based formulation is better than track-based formulation ?

- More efficient representation of “exclusivity constraint”
  - Responsible for about 80 % of total CNF clauses
- Track-based formulation exclusivity CNF clauses
  - Assuming there are 3 tracks per channel (W = 3)
  - **$X = [X_1\ X_0]$, $Y = [Y_1\ Y_0]$**
  - **X ¹ Y = (X_0 Ú X_1 Ú Y_0 Ú Y_1) Ù (Ø X_0 Ú X_1 Ú Ø Y_0 Ú Y_1) Ù**
    (X_0 Ú Ø X_1 Ú Y_0 Ú Ø Y_1)]
  - If there are T tracks per channel, each exclusivity constraint requires T CNF clauses consisting of $2*\log_2(T)$ literals

Why route-based formulation is better than track-based formulation ?

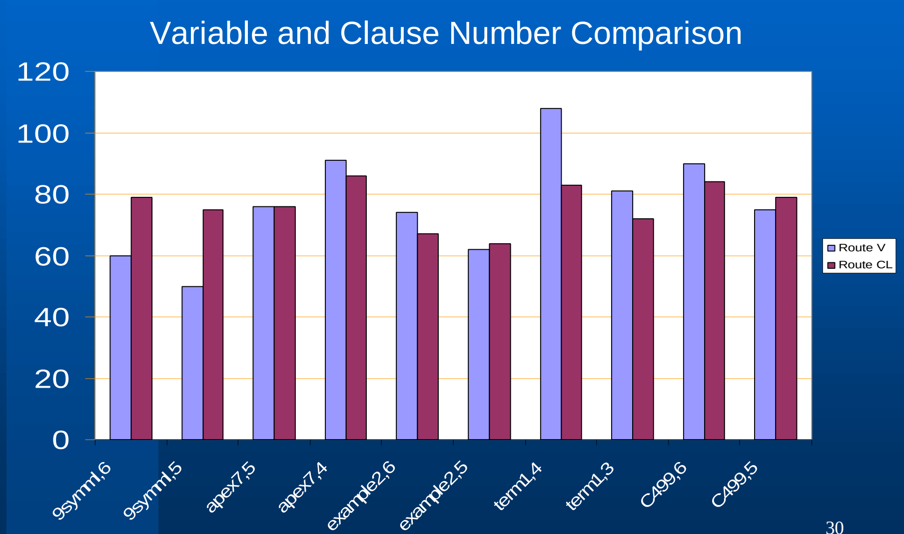
- More efficient representation of “exclusivity constraint”
  - Responsible for about 80 % of total CNF clauses
- Route-based formulation exclusivity CNF clauses
  - Each exclusivity constraint CNF clause is a NAND function
  - If there are T tracks per channel, each exclusivity constraint requires T CNF clauses consisting of 2 literals
- Thus, the final Routability Boolean function from route-based formulation is very close to 2-SAT CNF clauses.
- 2-SAT = P, 3-SAT = NP

Comparative Experimental Results

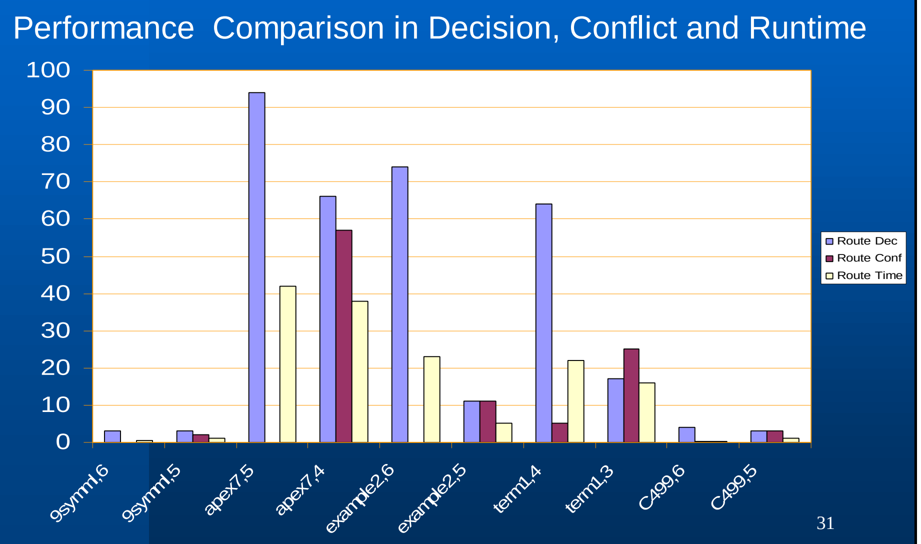
- Benchmark circuit suit
- Global routing result from VPR (Univ. of Toronto)

Circuit	CLB X x Y	CLB #	Net #	2pin Conn #
9symml	9 x 9	86	259	259
alu2	12 x 12	143	153	510
apex7	11 x 11	64	300	300
C499	10 x 10	74	312	312
C880	14 x 14	89	656	656
exmaple2	19 x 19	33	444	444
k2	19 x 19	99	1257	1257
term1	8 x 8	84	202	202
too_large	13 x 13	88	519	519
vda	15 x 15	92	722	722

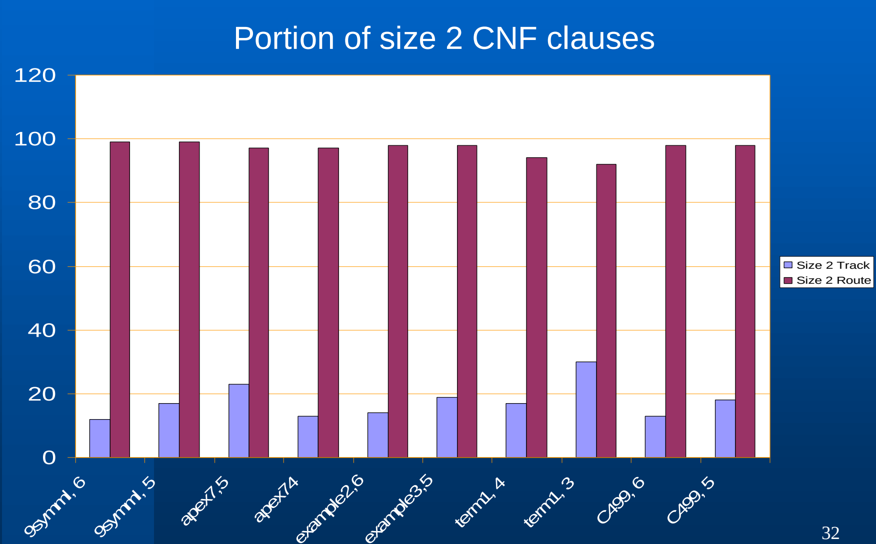
Comparative Experimental Results



Comparative Experimental Results



Comparative Experimental Results



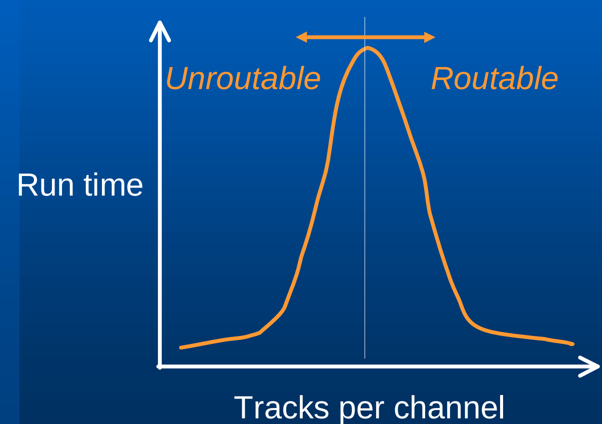
Comparative Experimental Results

- For other cases where “track”-based formulation couldn’t solve…….

Circuit	Variable	Clause	Decision	Conflict	Time
alu2, W=8	4080	83902	986	2	13.0
alu2, W=7	3570	73478	9014	8968	1191.6
C880, W=7	4592	61745	1143	81	13.8
C880, W=6	3936	53018	40327	39546	52364.7
k2, W=10	12570	338927	2902	51	186.1
k2, W=9	11313	305160	N.C	N.C	N.C
too_lrg, W=7	3633	50373	19	19	8.60
too_lrg, W=6	3114	43251	1184	1184	59.00
vda, W=8	5776	116527	24861	24861	6146.20
vda, W=7	5054	102547	22098	22098	12383.5

Comparative Experimental Results

- Typical Runtime graph



Comparative Experimental Results

- Solution Quality Comparison with other routers

Placement	VPR			SPLACE	
Global R	VPR				
Detailed R	Route	VPR	SEGA	SROUTE	FPR
9symml	6	5	6	7	9
alu2	8	7	9	8	10
apex7	5	4	6	6	9
exmaple2	6	6	6	7	13
term1	4	4	6	5	8
too_lrg	7	6	9	8	11
k2	10	9	11	11	17
vda	8	8	10	10	13
Total W	54	49	63	62	90

Conclusions and Future work

- Boolean Satisfiability-based FPGA routing
  - Simultaneous net-embedding (net-ordering independence)
  - Routability decision (estimation)
- FPGA Rerouting formulation via SAT
  - Track-based formulation
  - Route-based formulation
  - Comparative Experiment result
- More scalable formulation is needed
- Different application domains