Compatible Equivalence Checking of X-Valued Circuits

Authors: Yu-Neng Wang*, Yun-Rong Luo*, Po-Chun Chien*, Ping-Lun Wang,

Hao-Ren Wang, Wan-Hsuan Lin, Jie-Hong Roland Jiang and Chung-Yang Ric Huang

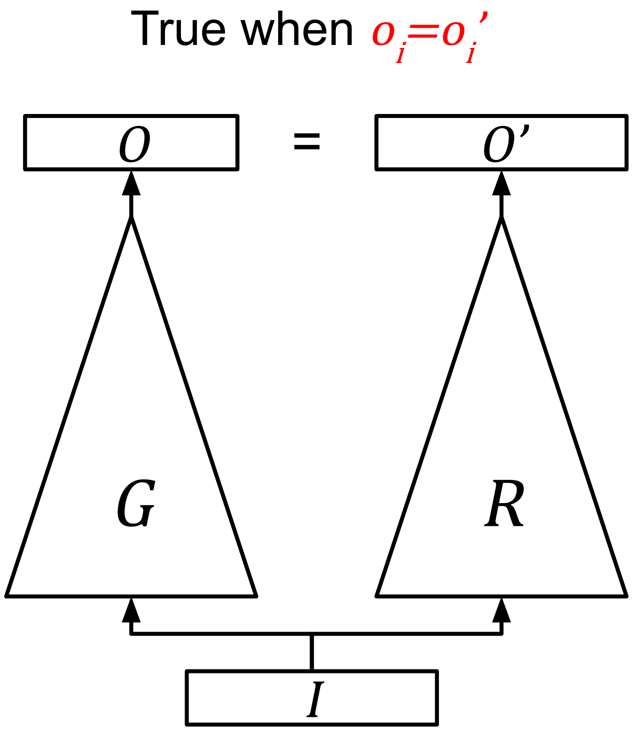
ALCom Lab

EE Dept./ Grad. Inst. of Electronics Eng.

National Taiwan University

*: equal contribution

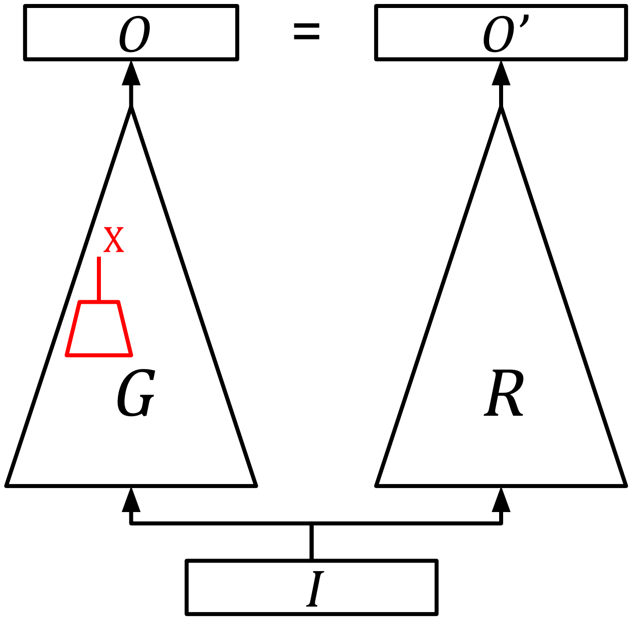
Combinational Equivalence Checking (CEC)



- I: Input pattern
- G: Golden netlist
- R: Revised netlist
- O/O': Output pattern of G/R under I

Compatible Equivalence Checking with X-value (XCEC)

True when $(o_i, o_i') =$ (0,0), (1,1), (x,0),(x,1),(x,x)



Definition 1. *Given two values* $\hat{a}, \hat{b} \in \mathbb{T}$, $\hat{a}$ *is* compatible equivalent *to* $\hat{b}$ *if* $(\hat{a}, \hat{b}) \in \{(0,0),(1,1),(\mathrm{x},0),(\mathrm{x},1),(\mathrm{x},\mathrm{x})\}$. *Otherwise,* $\hat{a}$ *is* not compatible equivalent *to* $\hat{b}$, *i.e.,* $(\hat{a}, \hat{b}) \in \{(0,1),(1,0),(0,\mathrm{x}),(1,\mathrm{x})\}$.

- Defined on ternary-valued logic
- Equivalence in golden circuit's care-space
- Asymmetric relation

X-Valued Circuits

- Primitive gates
  - *AND, NAND*
  - *OR, NOR*
  - *XOR, XNOR*
  - *NOT*
- Special gates
  - *DC*
  - *MUX*

a	0	1	x
NOT(a)	1	0	x

NOT(a)

b \ a	0	1	x
0	0	0	0
1	0	1	x
x	0	x	x

AND(a,b)

d \ c	0	1	x
0	0	1	x
1	x	x	x
x	x	x	x

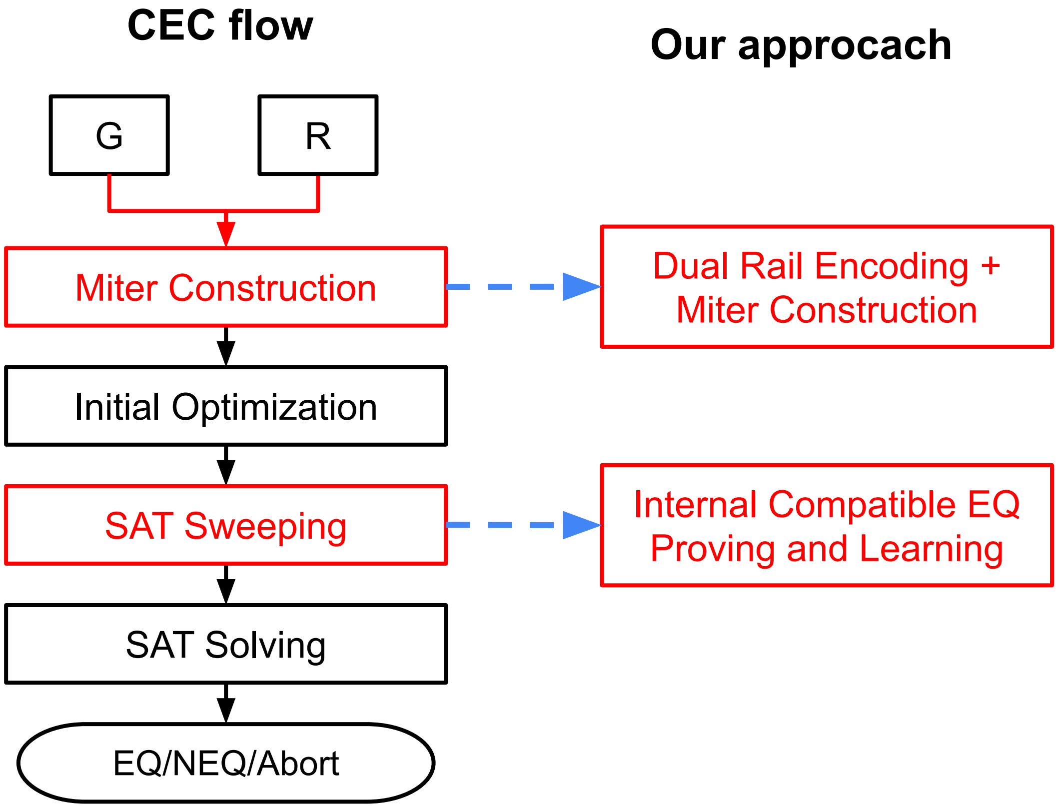
DC(c,d)

s=0				s=1				s=2			
b \ a	0	1	x	b \ a	0	1	x	b \ a	0	1	x
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	x	1	0	1	x	1	0	1	x
x	0	x	x	x	0	x	x	x	0	x	x

MUX(s,a,b)

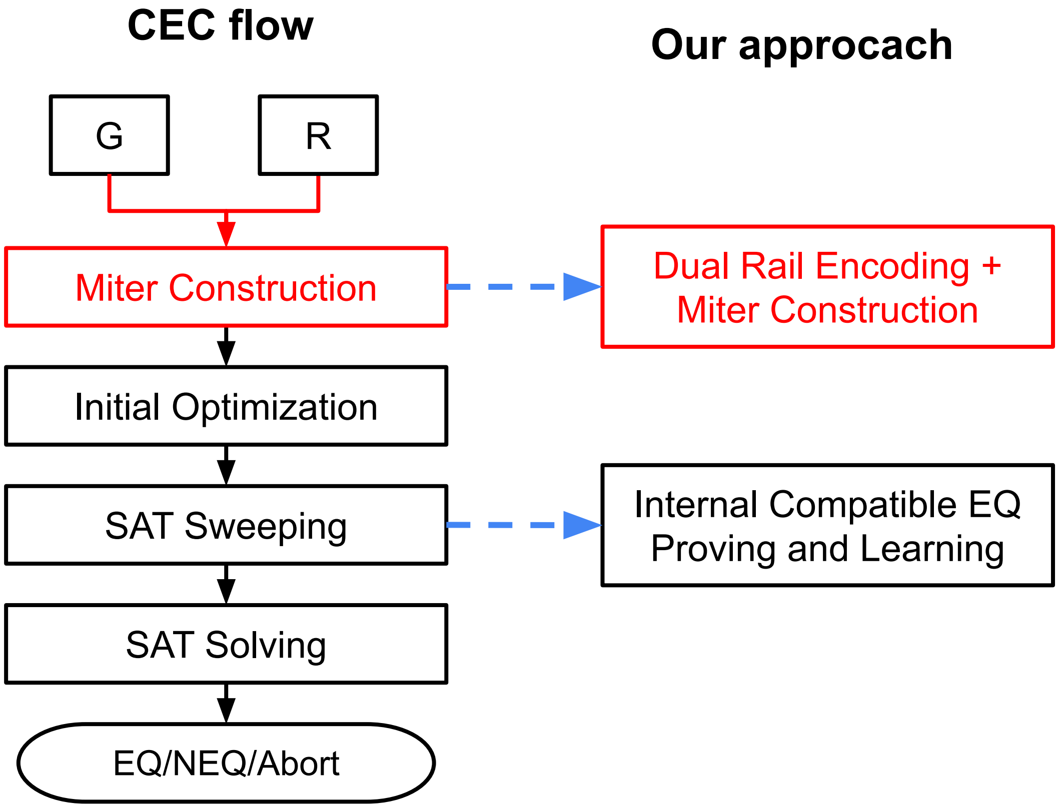
Proposed Algorithm Flow

From CEC to XCEC



- SAT solver is only applicable for binary logic
- Internal compatible equivalent pairs cannot be merged

Dual Rail Encoding



- Encoding choices
  - symmetric E_{sym}
  - X-preserving E_{xp}
  - one-hot E_{oh}

T	Encoded bits		
	$o^0\ o^1$		$o^0\ o^1\ o^2$
	E_{xp}	E_{sym}	E_{oh}
0	00	10	100
1	10	01	010
x	01, 11	00	001

o^1 bit is preserved to represent x

The superiority of X-preserving encoding: implication ability

We compare E_{xp} (X-preserving encoding) with E_{sym} (Symmetric encoding)

- E_{sym} is more succint than E_{xp} for most circuit primitive gates
- However, E_{xp} has stronger implication ability.

EQ	Ternary Loigc (T)	E_{xp}	E_{sym}
	(x,0) (x,1) (x,x)	(-1, - -)	(00, - -)
	(0,0)	(00, 00)	(10, 10)
	(1,1)	(10,10)	(01, 01)
NEQ	(1,0)	(10,00)	(01,-0)
	(1,x)	(-0,-1)	
	(0,x)		(10,0-)
	(0,1)	(00,10)	

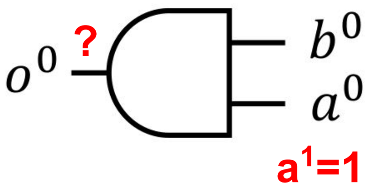
T	Encoded bits o^0 o^1	
	E_{xp}	E_{sym}
0	00	10
1	10	01
x	01, 11	00

E_{xp} can conclude EQ with 1 bit assignment while E_{sym} needs 2 bits assignment. E_{xp} has stronger implication ability.

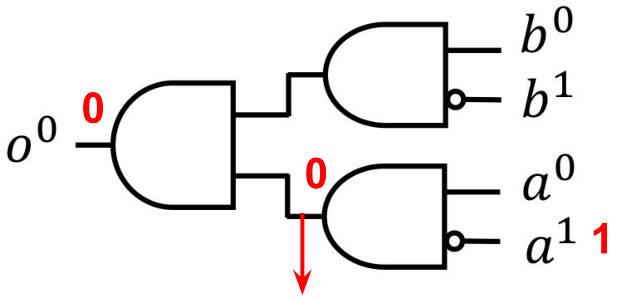
The superiority of X-preserving encoding: don't care property

Both a^0a^1 = 01, 11 represents x under Exp. When a^1= 1, the value of a^0 becomes don’t care.

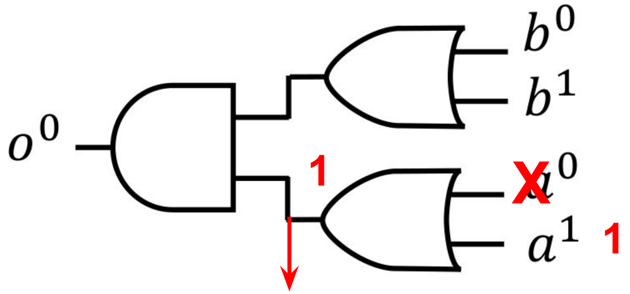
→ Replace a^0 to the controlling value/ non controlling value of o^0.



controlling value of AND: 0
non controlling value of AND: 1

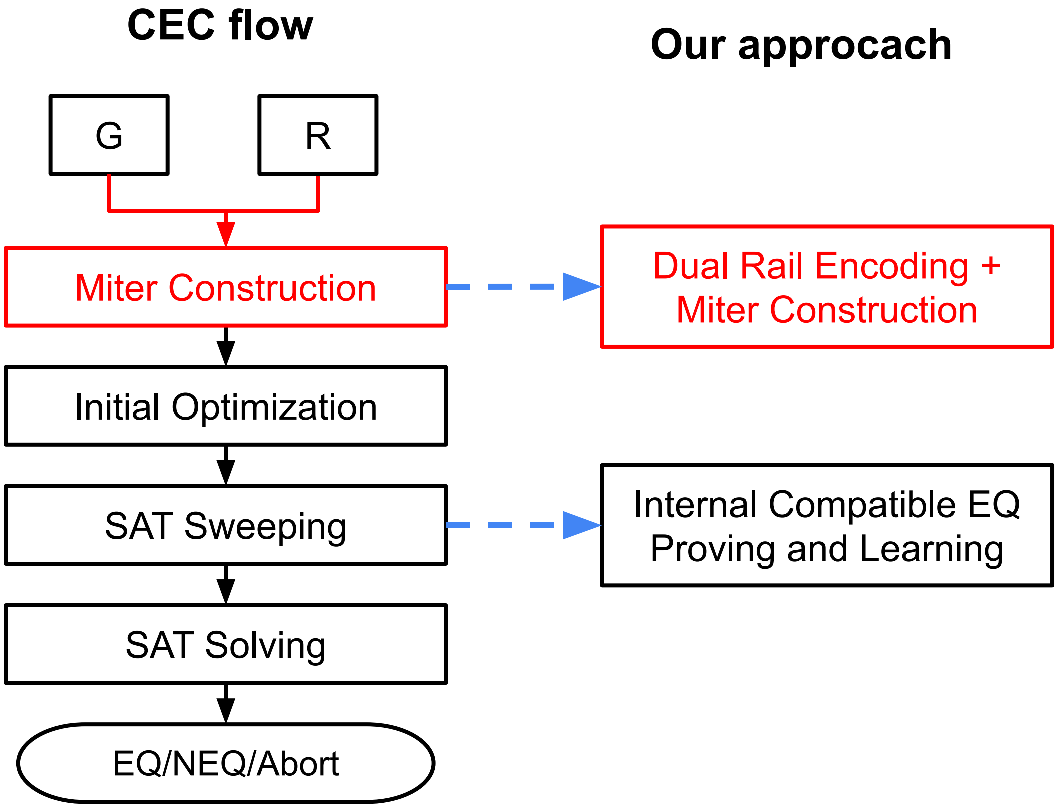


becomes 0 when a^1=1 (propagates the implication toward PO)



becomes 1 when a^1=1 (conditionally disable the fanin a^0 when a^1=1)

Dual Rail Encoding



- And gate under E_{xp}

$$o^0 o^1 = AND(a^0 a^1,\ b^0 b^1)$$

$$o^0 = a^0 b^0$$

$$o^1 = a^1 b^1 \vee a^1 b^0 \vee a^0 b^1$$

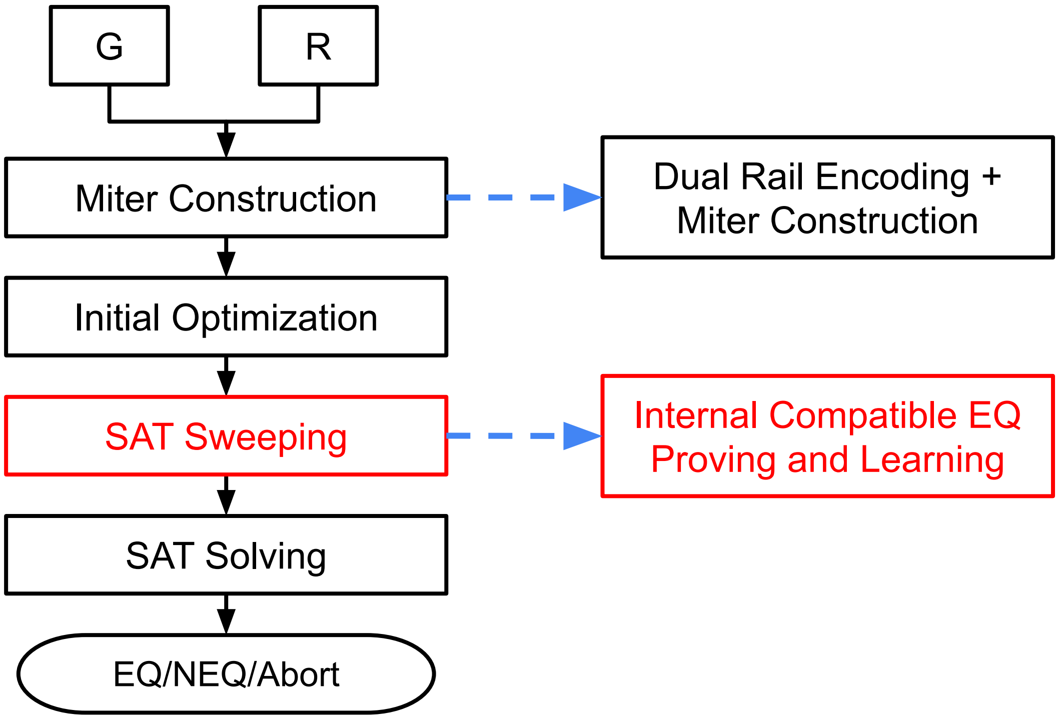
- Compatible EQ Miter under E_{xp}

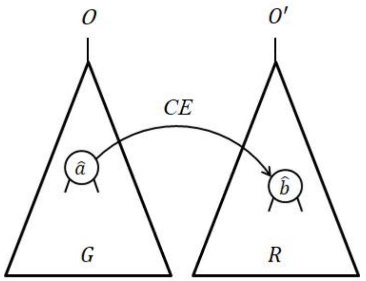
$$M = \bigvee_{i=1}^{n} (o^0_{g,i} \neg o^0_{r,i} \vee \neg o^0_{g,i} o^0_{r,i} \vee o^1_{r,i}) \neg o^1_{g,i}$$

Internal Compatible EQ(CE) Proving and Learning

CEC flow

Our approcach





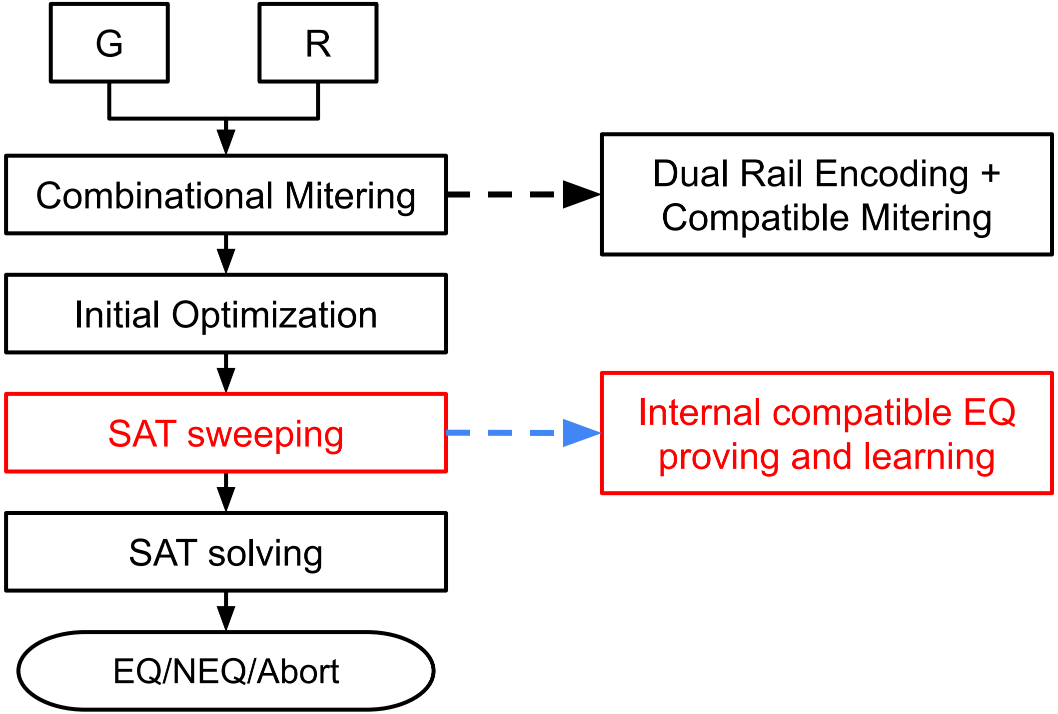
$$\mathrm{Exp}(\hat{a}) = (a^0, a^1)$$
$$\mathrm{Exp}(\hat{b}) = (b^0, b^1)$$

add to SAT instance

$$(a^1 \vee \neg b^1)$$
$$\wedge\, (a^1 \vee a^0 \vee \neg b^0)$$
$$\wedge\, (a^1 \vee \neg a^0 \vee b^0)$$

Internal Compatible EQ(CE) Proving and Learning

CEC flow **Our approach**



$$\text{Exp}(\hat{a}) = (a^0, a^1)$$
$$\text{Exp}(\hat{b}) = (b^0, b^1)$$

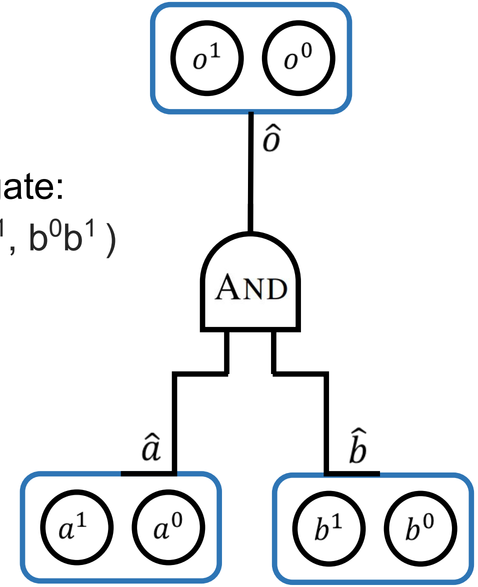
add to SAT instance

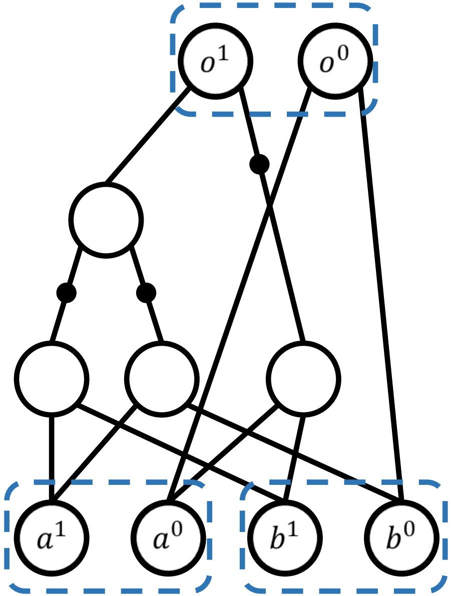
$$(a^1 \vee \neg b^1)$$
$$\wedge\, (a^1 \vee a^0 \vee \neg b^0)$$
$$\wedge\, (a^1 \vee \neg a^0 \vee b^0)$$

Circuit Representation

- Maintain: the high-level X-valued circuit and the low-level AIG.
- High-level circuit: the original ternary-valued circuit (consists of the primitive gates and constants)

high level AND gate:
$o^0 o^1 = \text{AND}(\ a^0 a^1,\ b^0 b^1\ )$



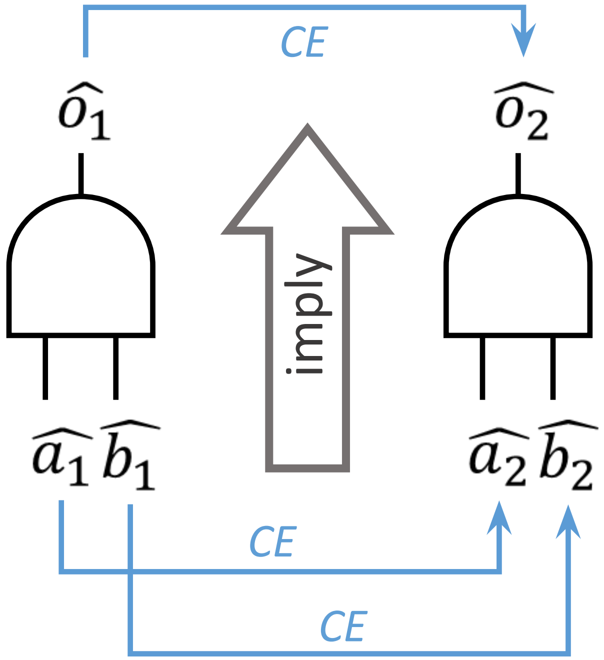


low level AND gate:
$o^0 = a^0\, b^0$
$o^1 = a^1\, b^1 \vee a^1\, b^0 \vee a^0\, b^1$

Propagating CE relation

Proposition: For a pair of ternary-valued signals $\hat{o}_1$ and $\hat{o}_2$ with $\hat{o}_1 = AND(\hat{a}_1, \hat{b}_1)$ and $\hat{o}_2 = AND(\hat{a}_2, \hat{b}_2)$, if $\hat{a}_1$ is CE to $\hat{a}_2$ and $\hat{b}_1$ is CE to $\hat{b}_2$, then $\hat{o}_1$ is CE to $\hat{o}_2$.

- Although CE pairs cannot be merged, we can use the proposition to propagate CE relation.
- Proving CE relation without time-consuming SAT solving



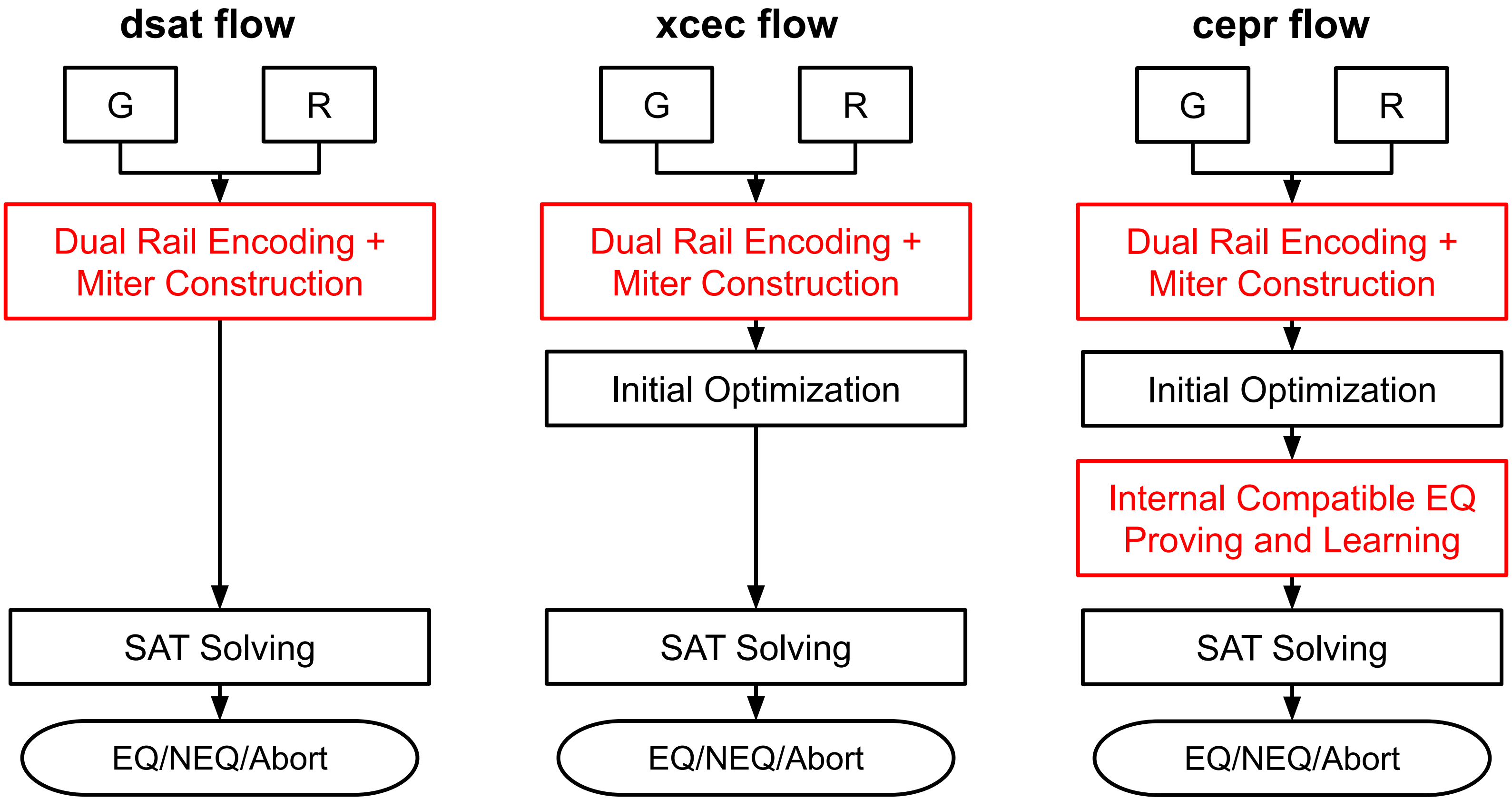
Experimental Results

Experimental Settings

- 2020 ICCAD CAD Contest Benchmark
  - 30 cases
    - 28 Industrial cases (23 EQ, 5 NEQ)
    - 2 Hard NEQ cases (excluded, no X-values)
  - 1,000 ~ 100,000 #Gates
  - Timeout limit 1800 secs
- Solver Setting
  - Berkeley ABC [1] (ABC 1.01 commit 5c8ee4a2c142d133afe4cbfe567b300fe4d040a8)
  - Incremental SAT solver: Glucose [2] (Glucose 3.0)
  - Final SAT solver: kissat [3] (kissat sc2020, target UNSAT)

[1] Brayton et al., 2010. [2] Audemard et al., 2009. [3] Biere et al., 2020.

Flow Comparison



Performance Evaluation

- Flow
  - xcec: encode → ABC circuit optimization → SAT solving
  - cepr: encode → ABC circuit optimization → CE proving and learning → SAT solving
- Encoding
  - x-preserving (E_{xp})
    - controlling value (E_{xp}^{c})
    - non-controlling value (E_{xp}^{nc})
  - symmetric (E_{sym})
- Baseline Method
  - Symmetric encoding
  - Other contestants
  - Conformal LEC

method	# solved cases			total time
	EQ	NEQ	total	
$xcec\text{-}E_{xp}^{nc}$	13	7	20	5625.25
$xcec\text{-}E_{xp}^{c}$	13	7	20	6305.45
$cepr\text{-}E_{xp}$	12	7	19	6645.13
$xcec\text{-}E_{xp}$	11	7	18	2600.02
3rd Place	11	7	18	2727.24
$xcec\text{-}E_{sym}$	11	7	18	4021.63
2nd Place	9	7	15	2157.75
LEC	6	5	11	2344.78

The superiority of X-preserving encoding: implication ability

- Under xcec flow, E_{xp} solves 18 cases in less total time than E_{sym}.

T	Encoded bits o^0 o^1	
	E_{xp}	E_{sym}
0	00	10
1	10	01
x	01, 11	00

method	# solved cases			total time
	EQ	NEQ	total	
$xcec\text{-}E_{xp}^{nc}$	13	7	20	5625.25
$xcec\text{-}E_{xp}^{c}$	13	7	20	6305.45
$cepr\text{-}E_{xp}$	12	7	19	6645.13
$xcec\text{-}E_{xp}$	11	7	18	2600.02
3rd Place	11	7	18	2727.24
$xcec\text{-}E_{sym}$	11	7	18	4021.63
2nd Place	9	7	15	2157.75
LEC	6	5	11	2344.78

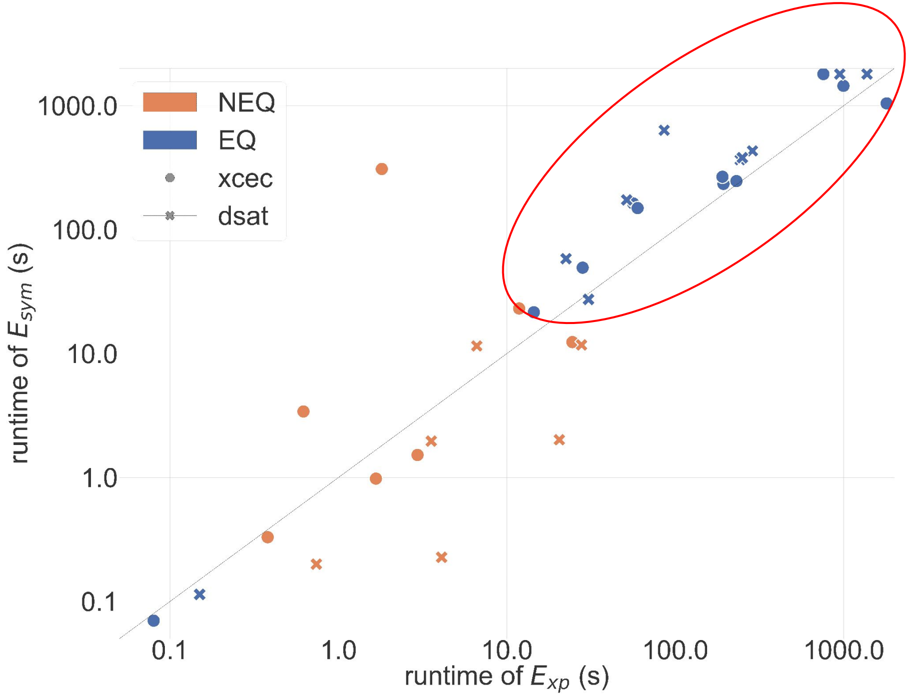
The superiority of X-preserving encoding: implication ability

Compare Exp and Esym under two flows:

- xcec: encode → ABC circuit optimization → SAT solving
- dsat: encode → SAT solving

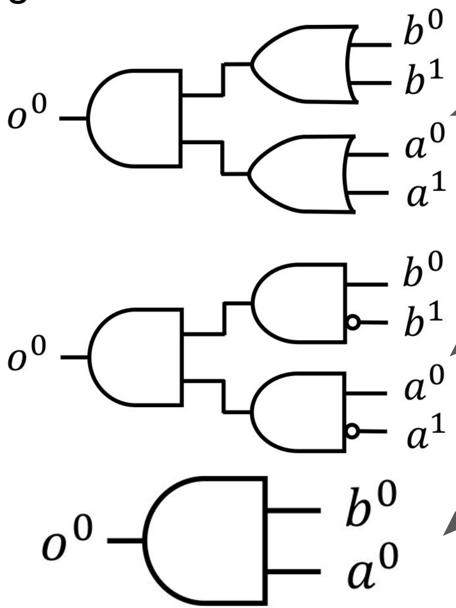
The superiority of Exp over Esym is independent of synthesis tool.

EQ	Ternary Loigc (T)	E_{xp}	E_{sym}
	(x,0) (x,1) (x,x)	(-1, - -)	(00, - -)
	(0,0)	(00, 00)	(10, 10)
	(1,1)	(10,10)	(01, 01)

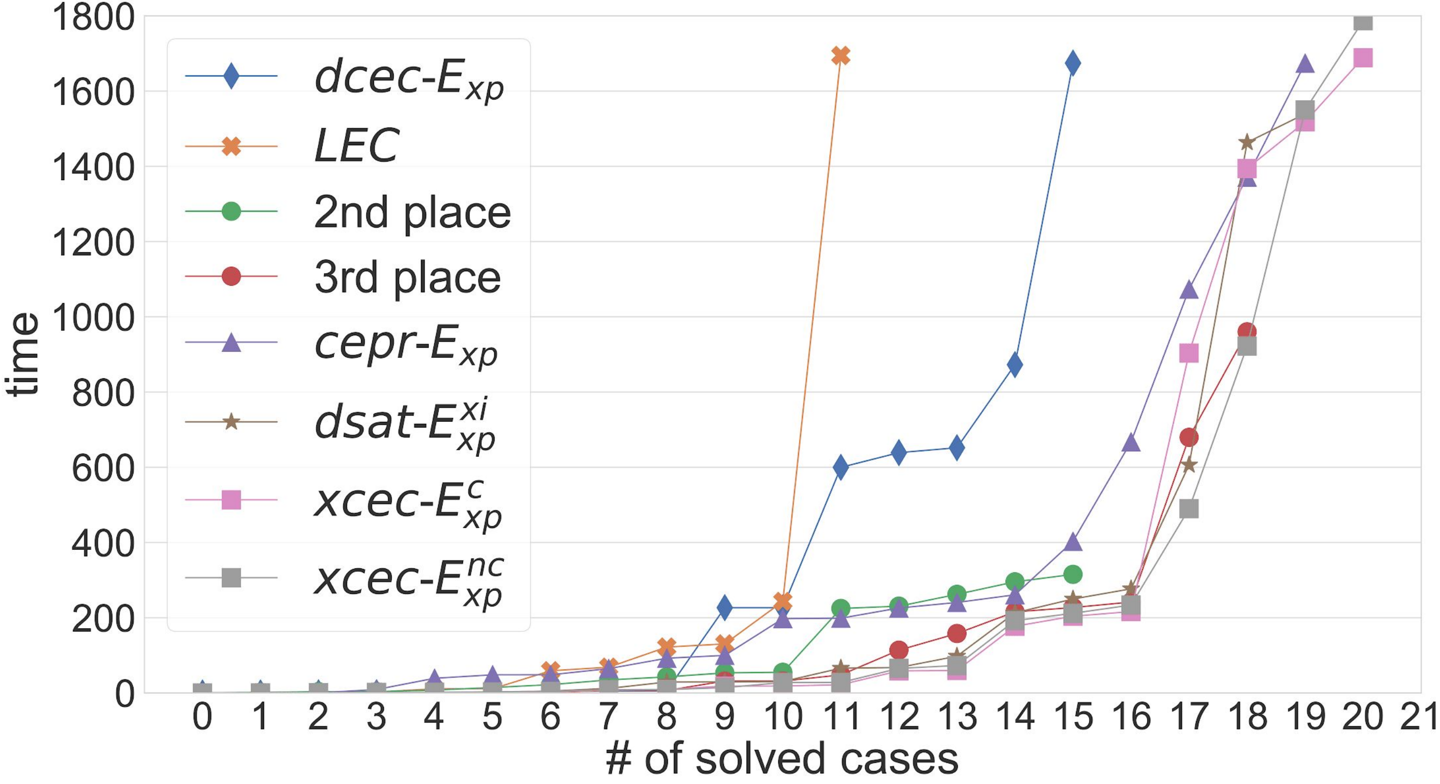


The superiority of X-preserving encoding: don't care property

$a^1 = 1$
→ the value of a^0 becomes don't care
→ replace a^0 to the controlling value/ non controlling value of o^0.

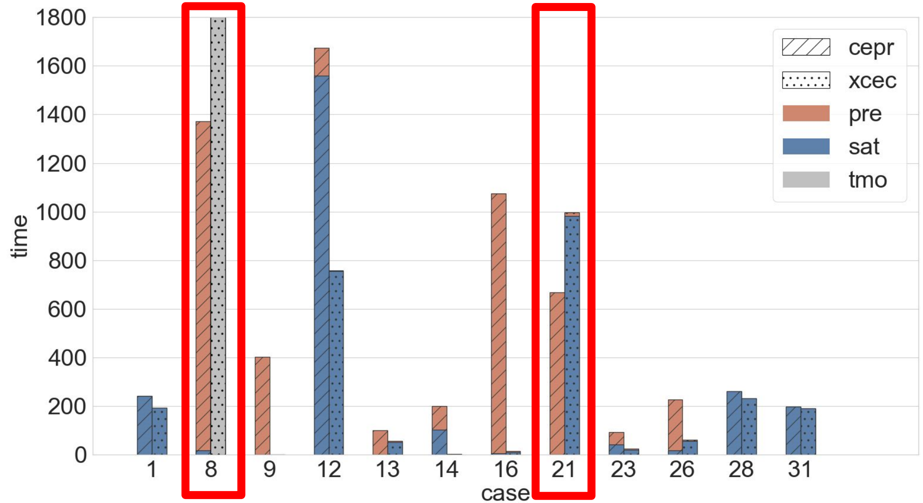


method	# solved cases			total time
	EQ	NEQ	total	
$xcec\text{-}E_{xp}^{nc}$	13	7	20	5625.25
$xcec\text{-}E_{xp}^{c}$	13	7	20	6305.45
$cepr\text{-}E_{xp}$	12	7	19	6645.13
$xcec\text{-}E_{xp}$	11	7	18	2600.02
3rd Place	11	7	18	2727.24
$xcec\text{-}E_{sym}$	11	7	18	4021.63
2nd Place	9	7	15	2157.75
LEC	6	5	11	2344.78



dcec-E_{xp}
LEC
2nd place
3rd place
cepr-E_{xp}
dsat-E_{xp}^{xi}
xcec-E_{xp}^{c}
xcec-E_{xp}^{nc}
time
1800
1600
1400
1200
1000
800
600
400
200
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
# of solved cases

Internal CE Learning Improves Final SAT Solving



method	# solved cases			total time
	EQ	NEQ	total	
$xcec\text{-}E^{nc}_{xp}$	13	7	20	5625.25
$xcec\text{-}E^{c}_{xp}$	13	7	20	6305.45
$cepr\text{-}E_{xp}$	12	7	19	6645.13
$xcec\text{-}E_{xp}$	11	7	18	2600.02
3rd Place	11	7	18	2727.24
$xcec\text{-}E_{sym}$	11	7	18	4021.63
2nd Place	9	7	15	2157.75
LEC	6	5	11	2344.78

Conclusion

- With stronger implication ability, x-preserving encoding outperforms traditional symmetric encoding.

- Using don’t-care property further improves the performance of x-preserving encoding.

- Learned clauses from internal CE relation speed up final SAT solving.

Thank you for your listening

Acknowledgement

- Cadence …
- 2nd, 3rd Place ...