Logic Synthesis of Recombinase-based Genetic Circuits

Jie-Hong Roland Jiang

Department of Electrical Engineering
Graduate Institute of Electronics Engineering
National Taiwan University

Design Automation for Wetware!

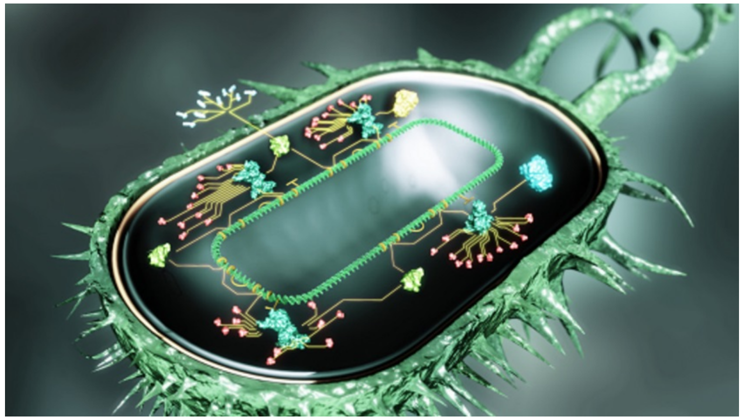
- Hardware
  - Data representation
    - Voltage
  - Signal
    - Wires
    - “Unlimited” signals
  - High predictability
    - Well-controlled electrical environment
  - Design principle
    - Mostly digital
    - Mostly synchronous

- Wetware
  - Data representation
    - Concentration
  - Signal
    - Molecular species
    - Limited signals
  - Low predictability
    - Noisy biochemical environment
  - Design principle
    - Analog + digital
    - Mostly asynchronous

Outline

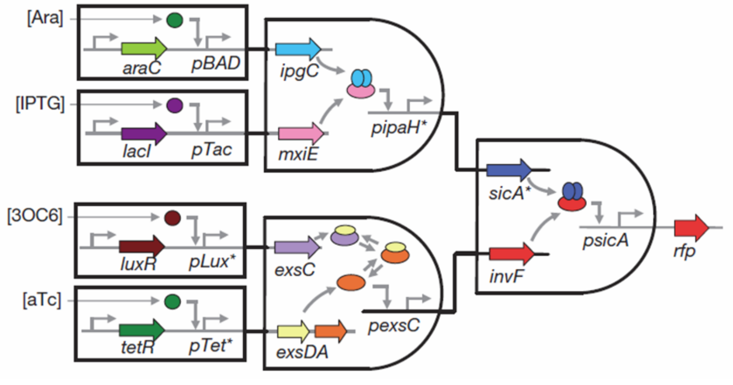
Systems and Synthetic Biology

- Biotechnology has promising applications in health, medicine, environment, food, energy, etc.
- Learn circuit design principles from nature
  - Complex behaviors of bacteria, say, emerge from fundamental biochemical reactions
- Engineer circuit components and systems from known biochemical parts
  - Bio-design automation


Source: openbiolabs.org

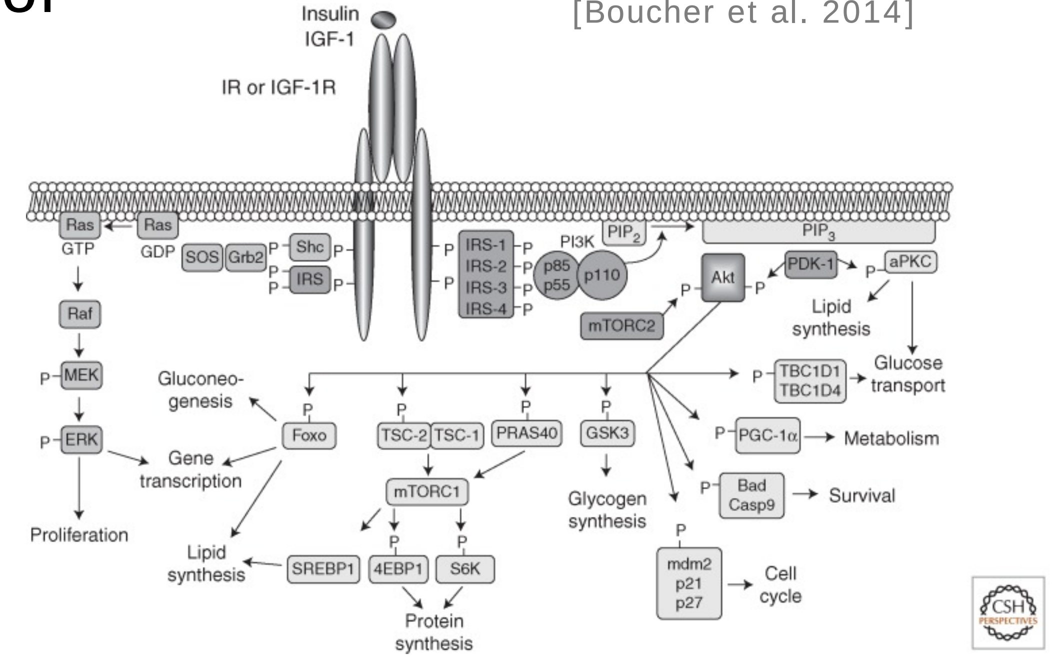
Why Logic Design in Biology ?

- Computation in living cells
  - Sensing / diagnosis
  - Decision making
  - Response / control



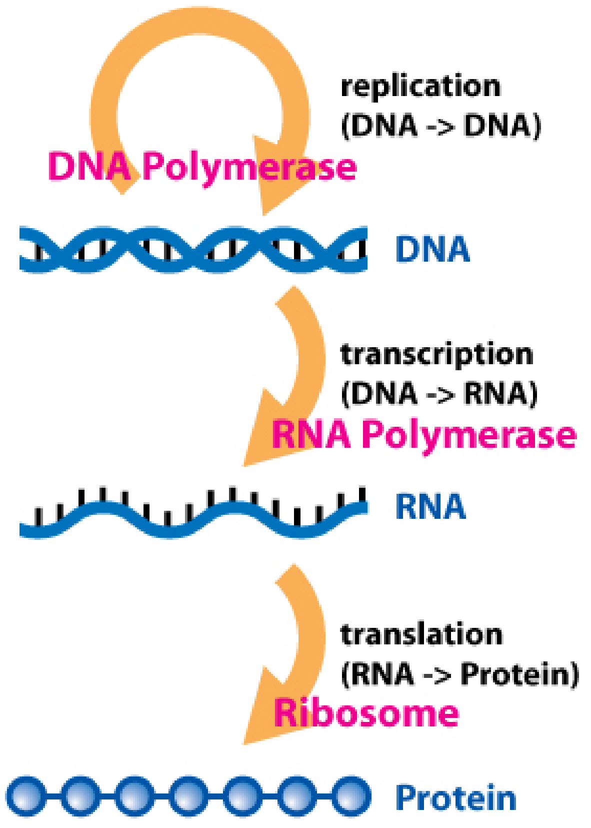
Moon et al. 2012

Insulin signaling pathways [Boucher et al. 2014]

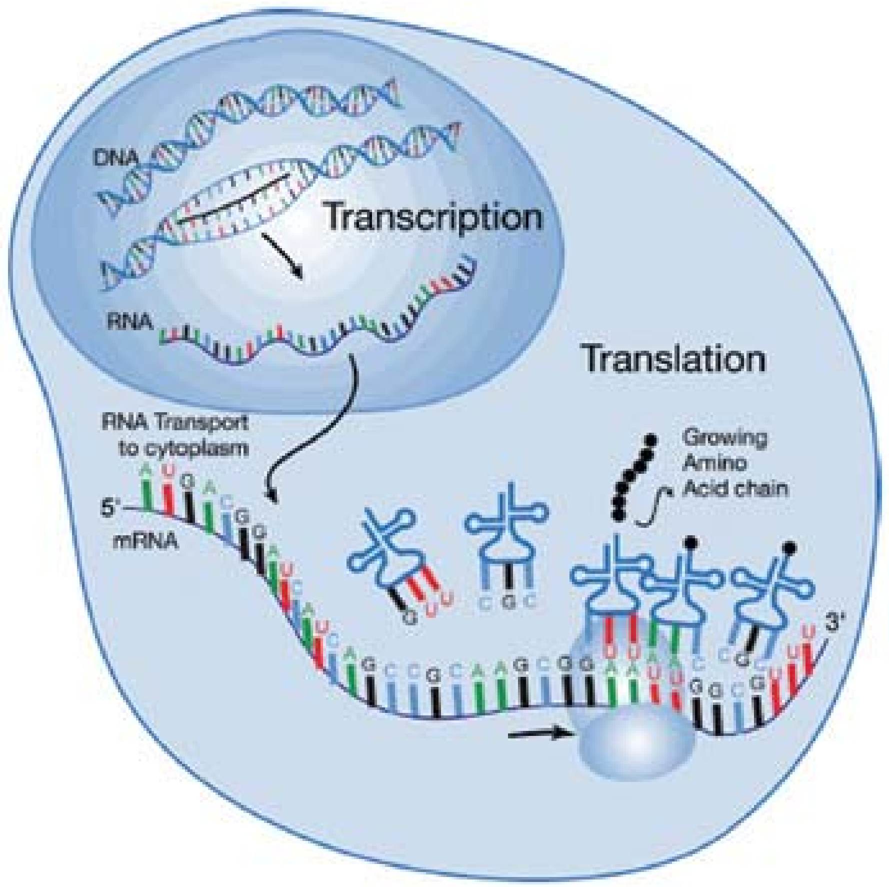


Molecular Biology 101

- Central dogma of molecular biology



Source: Wikipedia



Gene Transcription

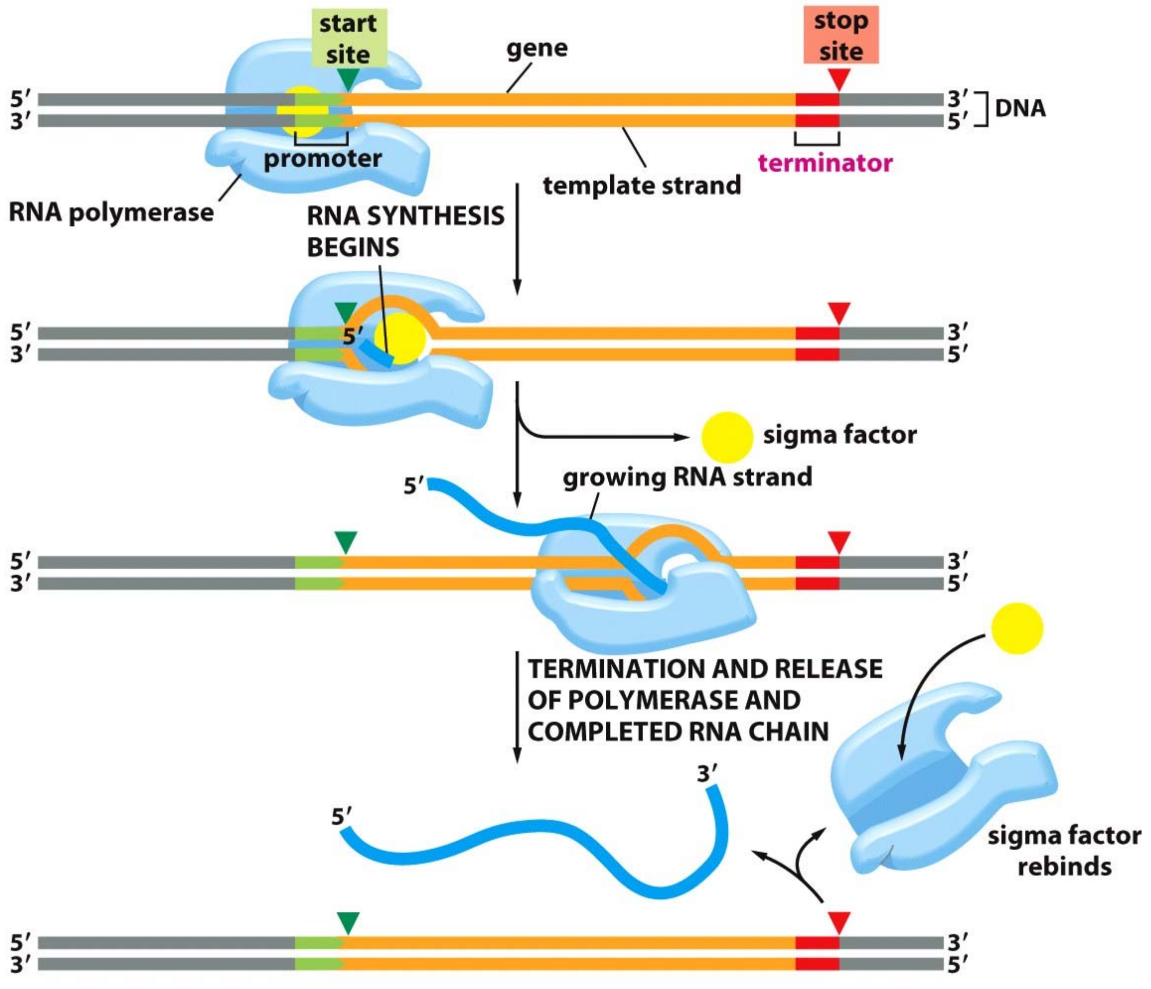
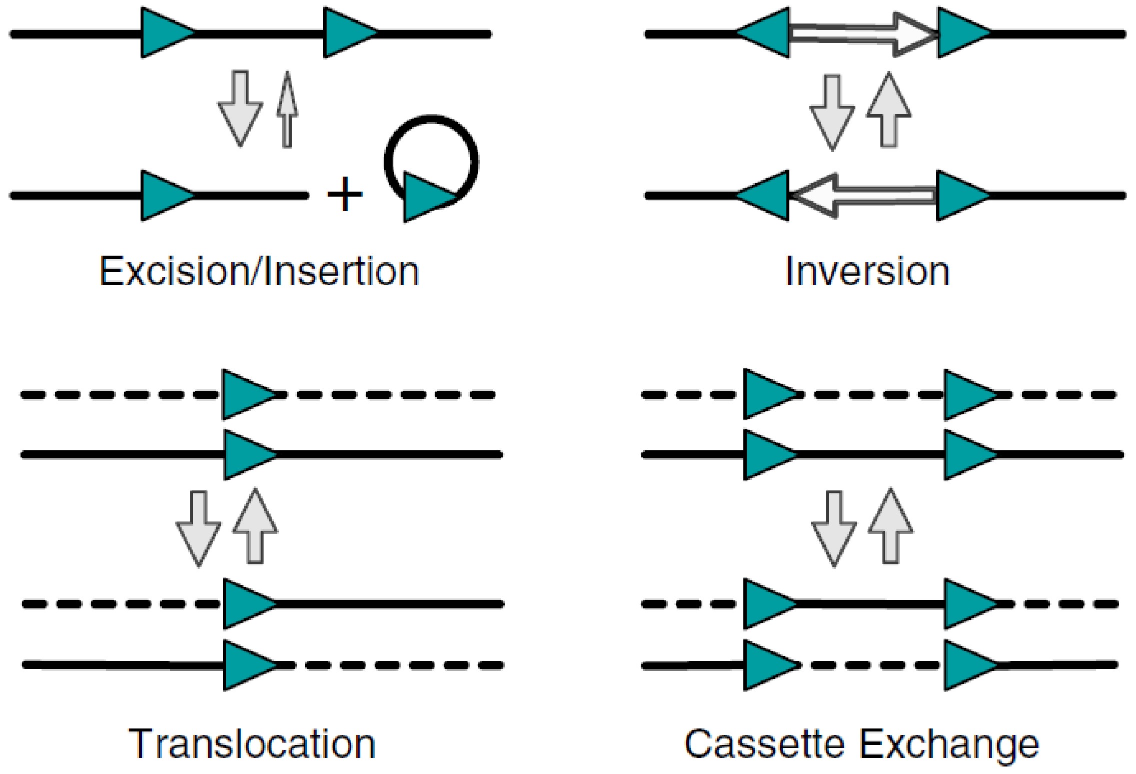


Figure 7-9 Essential Cell Biology 3/e (© Garland Science 2010)

Genetic Recombination

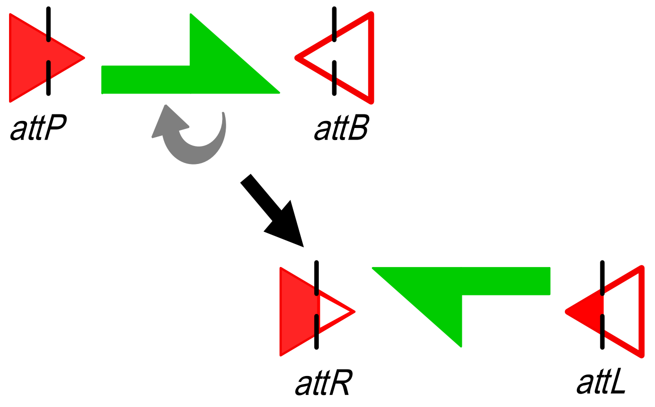
- A **recombinase** is an enzyme that achieves one of the following types of genetic recombination



[Nern et al. 2011]

Site-Specific Recombination

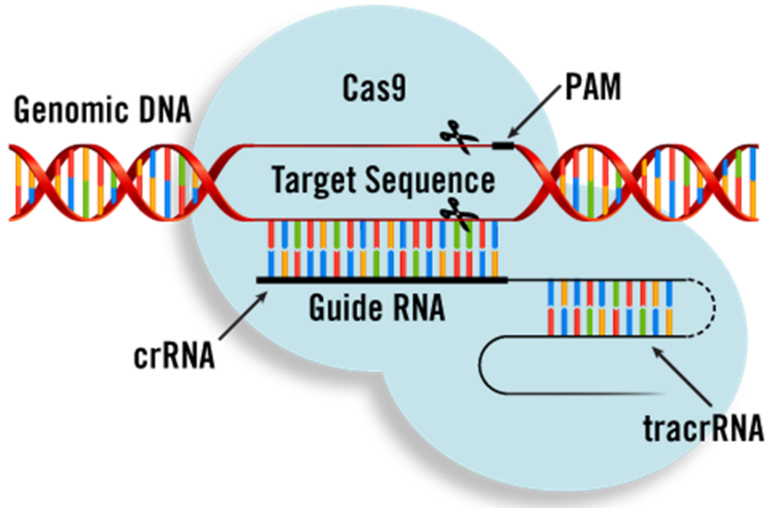
- Site-specific recombinase for inversion (and excision)



  - Irreversible (mostly)
  - Long-term memory effect cross generations

Other Genome Editing Method

- CRISPR/Cas9 systems



Source: mobitec.com

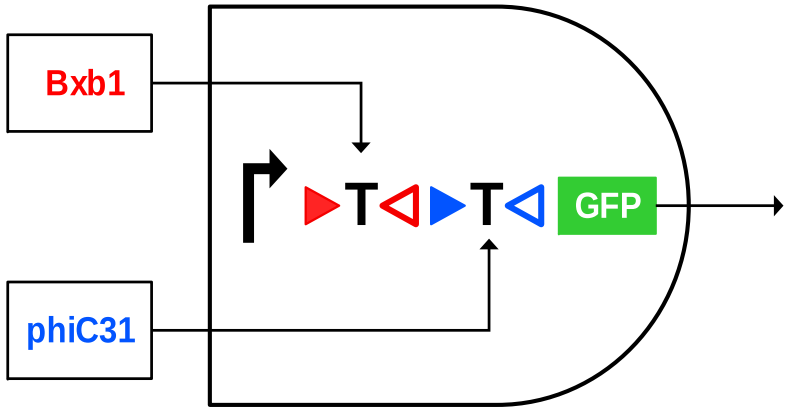
  - Excision/insertion in genome editing
  - Activation/inhibition in gene expression regulation (defective form of Cas9)

Outline

- Introduction
- **Formalism**
- Genetic Circuit Synthesis
- Genetic Circuit Optimization
- Summary and future work

Logic Design with Recombinase

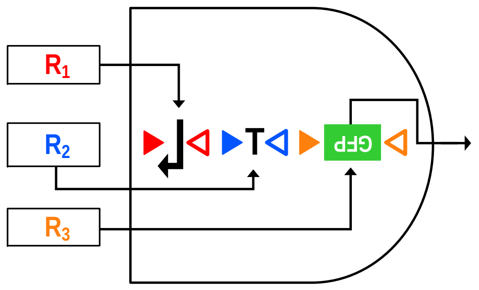
↱	promoter
T	transcription terminator
▶ ◁	targeting sites
🟩 GFP	green fluorescent protein



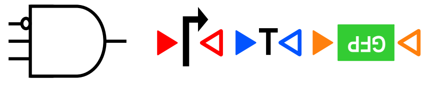
Bxb1	phiC31	GFP
0	0	0
1	0	0
0	1	0
1	1	1

Recombinase-based Logic Gates

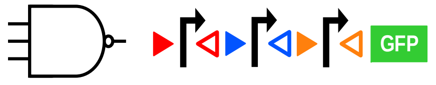
(a)



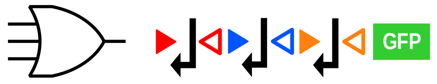
(b)



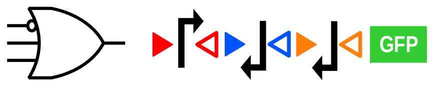
(c)



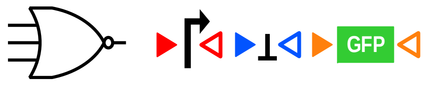
(d)



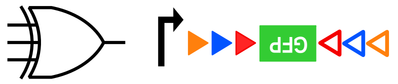
(e)



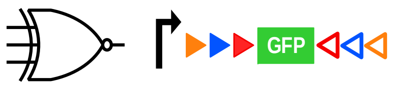
(f)



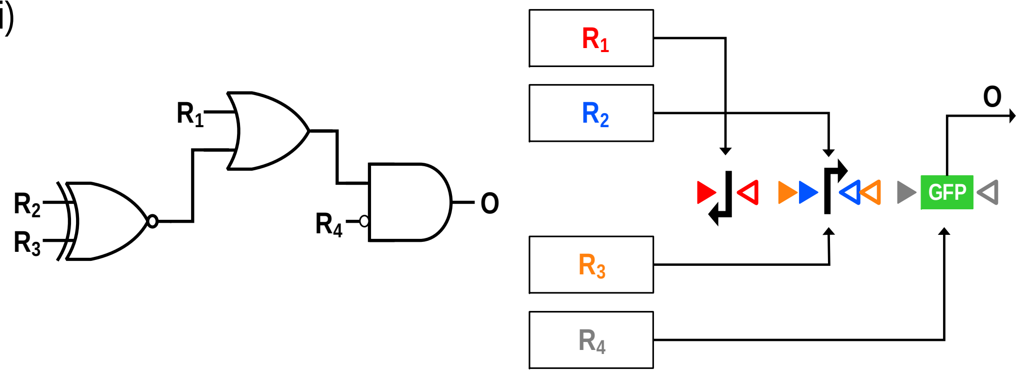
(g)



(h)



(i)



$$O = (R_1 + \overline{R_2} \oplus R_3)\overline{R_4}$$

Recombinase Logic Gate: Syntax

- Atomic terms

$$\langle \textit{atomic term} \rangle ::= P \,|\, d \,|\, T \,|\, \perp \,|\, G \,|\, \reflectbox{G}$$

- Well-formed sequence

$$\langle \textit{wfs} \rangle ::= \langle \textit{atomic term} \rangle \mid \{\langle \textit{wfs} \rangle\}_{r_i} \mid \langle \textit{wfs} \rangle \langle \textit{wfs} \rangle$$

  - E.g.,

$\boxed{\{T\}_{r_1}G}$ $\boxed{d\{T\}_{r_1}G}$ $\boxed{\{\{d\{T\}_{r_1}\}_{r_2}\}_{r_3}G}$ $\boxed{\{\{\{P\}_{r_5}\{\perp\}_{r_4}\}_{r_6}\{\{d\{T\}_{r_1}\}_{r_2}\}_{r_3}\}_{r_7}G}$

$\boxed{\{P\perp\}_{r_1}G}$ $\boxed{\{\{d\}_{r_1}\}_{r_2}G}$ $\boxed{\{T\{\perp\}_{r_1}\{P\}_{r_2}d\}_{r_3}G}$

Recombinase Logic Gate: Semantics

- Reduction rules

$$\sigma_1\{T\}_r\sigma_2 G \equiv \begin{cases} \sigma_2 G, & \text{for } R = 0 \\ \sigma_1\sigma_2 G, & \text{for } R = 1 \end{cases}$$

$$\sigma_1\{L\}_r\sigma_2 G \equiv \begin{cases} \sigma_1\sigma_2 G, & \text{for } R = 0 \\ \sigma_2 G, & \text{for } R = 1 \end{cases}$$

$$\sigma_1\{P\}_r\sigma_2 G \equiv \begin{cases} P\sigma_2 G, & \text{for } R = 0 \\ \sigma_1\sigma_2 G, & \text{for } R = 1 \end{cases}$$

$$\sigma_1\{d\}_r\sigma_2 G \equiv \begin{cases} \sigma_1\sigma_2 G, & \text{for } R = 0 \\ P\sigma_2 G, & \text{for } R = 1 \end{cases}$$

Recombinase Logic Gate: Semantics

- Ω-operator on a well-formed sequence

component C	**operator** $\Omega[\sigma C]$
T	$0\cdot(\Omega[\sigma])$
P	$1+(\Omega[\sigma])$
$\{T\}_r$	$R\cdot(\Omega[\sigma])$
$\{P\}_r$	$\overline{R}+(\Omega[\sigma])$
$\perp$	$1\cdot(\Omega[\sigma])$
d	$0+(\Omega[\sigma])$
$\{\perp\}_r$	$\overline{R}\cdot(\Omega[\sigma])$
$\{d\}_r$	$R+(\Omega[\sigma])$

$$\Omega[\emptyset]=0$$

Recombinase Logic Gate: Semantics

- Example

$$\begin{tikzcd}
\{T\}_{r_5}\{P\}_{r_4}\{\perp\}_{r_3}\{d\}_{r_2}\{T\}_{r_1} & R_1 \arrow[r, "0"] \arrow[d, "1"'] & 0 & \\
\{T\}_{r_5}\{P\}_{r_4}\{\perp\}_{r_3}\{d\}_{r_2} & R_2 \arrow[r, "1"] \arrow[d, "0"'] & 1 & P \\
\{T\}_{r_5}\{P\}_{r_4}\{\perp\}_{r_3} & R_3 \arrow[r, "1"] \arrow[d, "0"'] & 0 & \\
\{T\}_{r_5}\{P\}_{r_4} & R_4 \arrow[r, "0"] \arrow[d, "1"'] & 1 & P \\
\{T\}_{r_5} & R_5 \arrow[r, "0"] \arrow[d, "1"'] & 0 & \\
 & 0 & &
\end{tikzcd}$$

$$\begin{aligned}
&\Omega[\overbrace{\{T\}_{r_5}\{P\}_{r_4}\{\perp\}_{r_3}\{d\}_{r_2}}^{\sigma}\overbrace{\{T\}_{r_1}}^{C}] \\
=\ & R_1(\Omega[\overbrace{\{T\}_{r_5}\{P\}_{r_4}\{\perp\}_{r_3}}^{\sigma}\overbrace{\{d\}_{r_2}}^{C})\\
=\ & R_1(R_2+(\Omega[\{T\}_{r_5}\{P\}_{r_4}\{\perp\}_{r_3}]))\\
=\ & R_1(R_2+(\overline{R_3}(\Omega[\{T\}_{r_5}\{P\}_{r_4}])))\\
=\ & R_1(R_2+(\overline{R_3}(\overline{R_4}+(\Omega[\{T\}_{r_5}]))))\\
=\ & R_1(R_2+(\overline{R_3}(\overline{R_4}+(R_5(\Omega[\perp])))))\\
=\ & R_1(R_2+(\overline{R_3}(\overline{R_4}+(R_5(0)))))\\
=\ & R_1(R_2+(\overline{R_3}\,\overline{R_4})).
\end{aligned}$$

component C	operator $\Omega[\sigma C]$
T	$0\cdot(\Omega[\sigma])$
P	$1+(\Omega[\sigma])$
$\{T\}_r$	$R\cdot(\Omega[\sigma])$
$\{P\}_r$	$\overline{R}+(\Omega[\sigma])$
$\perp$	$1\cdot(\Omega[\sigma])$
d	$0+(\Omega[\sigma])$
$\{\perp\}_r$	$\overline{R}\cdot(\Omega[\sigma])$
$\{d\}_r$	$R+(\Omega[\sigma])$

Recombinase Logic Gate: Semantics

- $\Omega[\{\sigma\}_r] = \overline{R} \cdot \Omega[\sigma] + R \cdot \Omega[\reflectbox{σ}]$

  - Example

$$
\begin{aligned}
& \Omega[\{\{T\}_{r_5}\{P\}_{r_4}\{\reflectbox{L}\}_{r_3}\{d\}_{r_2}\{T\}_{r_1}\}_{r_6}] \\
= \; & \overline{R_6}\Omega[\{T\}_{r_5}\{P\}_{r_4}\{\reflectbox{L}\}_{r_3}\{d\}_{r_2}\{T\}_{r_1}] + R_6\Omega[\{\reflectbox{L}\}_{r_1}\{P\}_{r_2}\{T\}_{r_3}\{d\}_{r_4}\{\reflectbox{L}\}_{r_5}] \\
= \; & \overline{R_6}(R_1(R_2 + (\overline{R_3}(\overline{R_4} + (R_5(0)))))) + R_6(\overline{R_5}(R_4 + (R_3(\overline{R_2} + (\overline{R_1}(0)))))) \\
= \; & \overline{R_6}\,R_1(R_2 + \overline{R_3}\,\overline{R_4}) + R_6\overline{R_5}(R_4 + R_3\overline{R_2}).
\end{aligned}
$$

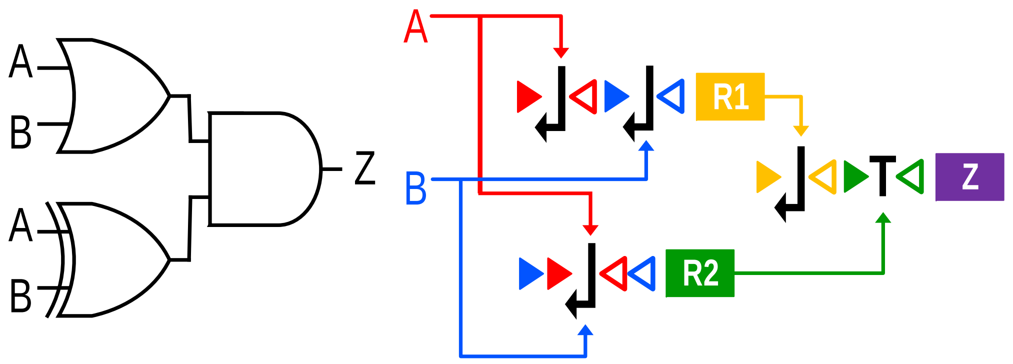
$$
\begin{aligned}
& \Omega[\{\{P\}_{r_4}\{\{\reflectbox{L}\}_{r_3}\{d\}_{r_2}\}_{r_5}\{T\}_{r_1}\}_{r_6}] \\
= \; & \cdots \\
= \; & \overline{R_6}\,R_1(\overline{R_5}(R_2 + \overline{R_3}\,\overline{R_4}) + R_5(R_3(\overline{R_2} + \overline{R_4}))) + R_6(R_4 + \overline{R_5}\,R_3\overline{R_2} + R_5R_2).
\end{aligned}
$$

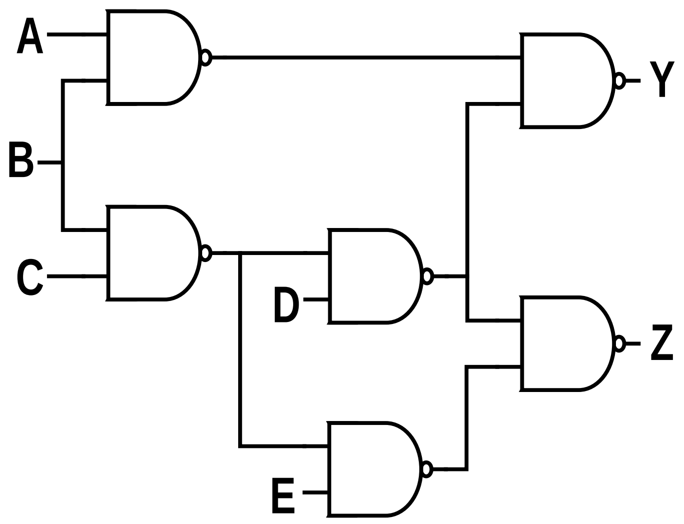
Outline

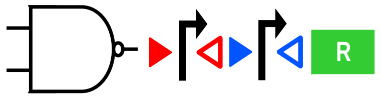
- Introduction
- Formalism
- **Genetic Circuit Synthesis**
- Genetic Circuit Optimization
- Summary and future work

Multi-level Recombinase Circuits

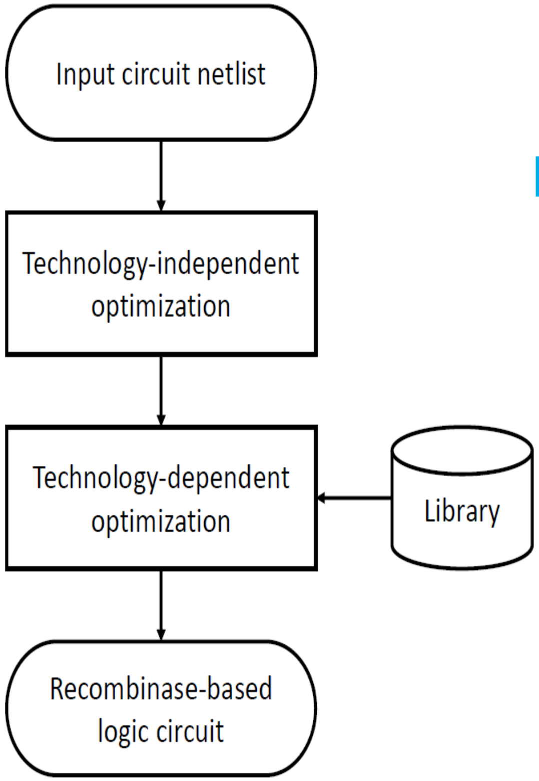
- Example







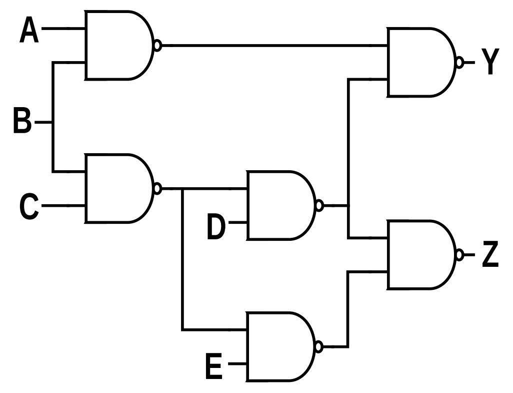
Logic Synthesis of Recombinase Circuits

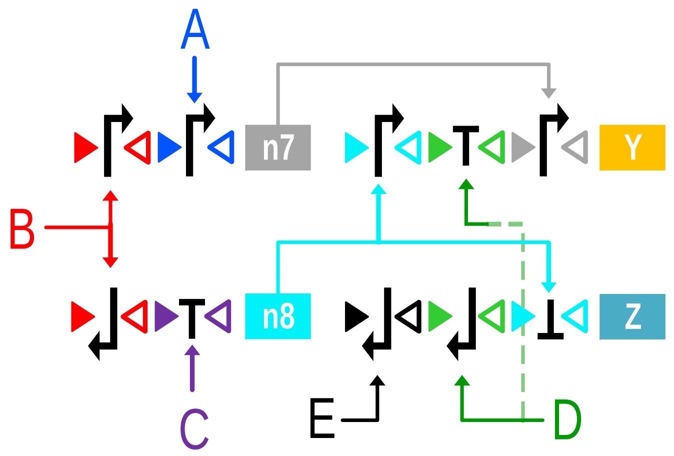

Input circuit netlist
Technology-independent optimization
Technology-dependent optimization
Library
Recombinase-based logic circuit

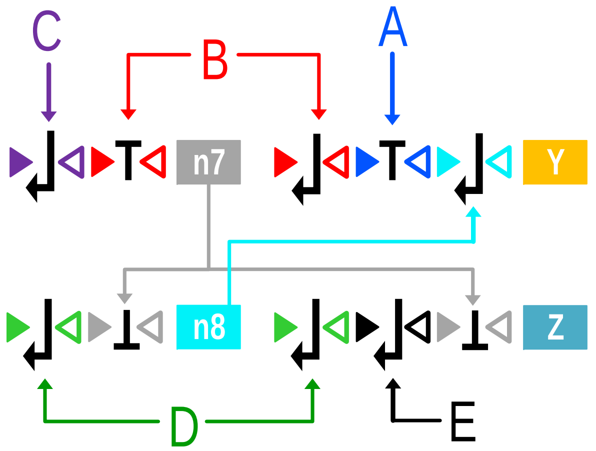
basic logic gates with up to three inputs

NAME	AREA	FUNCTION
c1_1	1	O = a
c1_2	1	O = !a
c2_1	2	O = a+(b)
c2_2	2	O = a+(!b)
c2_3	2	O = !a+(b)
c2_4	2	O = !a+(!b)
c2_5	2	O = a*(b)
c2_6	2	O = a*(!b)
c2_7	2	O = !a*(b)
c2_8	2	O = !a*(!b)
c3_1	3	O = a+(b+(c))
c3_2	3	O = a+(b+(!c))
c3_3	3	O = a+(!b+(c))
c3_4	3	O = a+(!b+(!c))
zero	0	O = CONST0
c3_5	3	O = a+(b*(c))
c3_6	3	O = a+(b*(!c))
c3_7	3	O = a+(!b*(c))
c3_8	3	O = a+(!b*(!c))
c3_9	3	O = !a+(b+(c))
c3_10	3	O = !a+(b+(!c))
c3_11	3	O = !a+(!b+(c))
c3_12	3	O = !a+(!b+(!c))
c3_13	3	O = !a+(b*(c))
c3_14	3	O = !a+(b*(!c))
c3_15	3	O = !a+(!b*(c))
c3_16	3	O = !a+(!b*(!c))
c3_17	3	O = a*(b+(c))
c3_18	3	O = a*(b+(!c))
one	1	O = CONST1
c3_19	3	O = a*(!b+(c))
c3_20	3	O = a*(!b+(!c))
c3_21	3	O = a*(b*(c))
c3_22	3	O = a*(b*(!c))
c3_23	3	O = a*(!b*(c))
c3_24	3	O = a*(!b*(!c))
c3_25	3	O = !a*(b+(c))
c3_26	3	O = !a*(b+(!c))
c3_27	3	O = !a*(!b+(c))
c3_28	3	O = !a*(!b+(!c))
c3_29	3	O = !a*(b*(c))
c3_30	3	O = !a*(b*(!c))
c3_31	3	O = !a*(!b*(c))
c3_32	3	O = !a*(!b*(!c))

Logic Synthesis of Recombinase Circuits





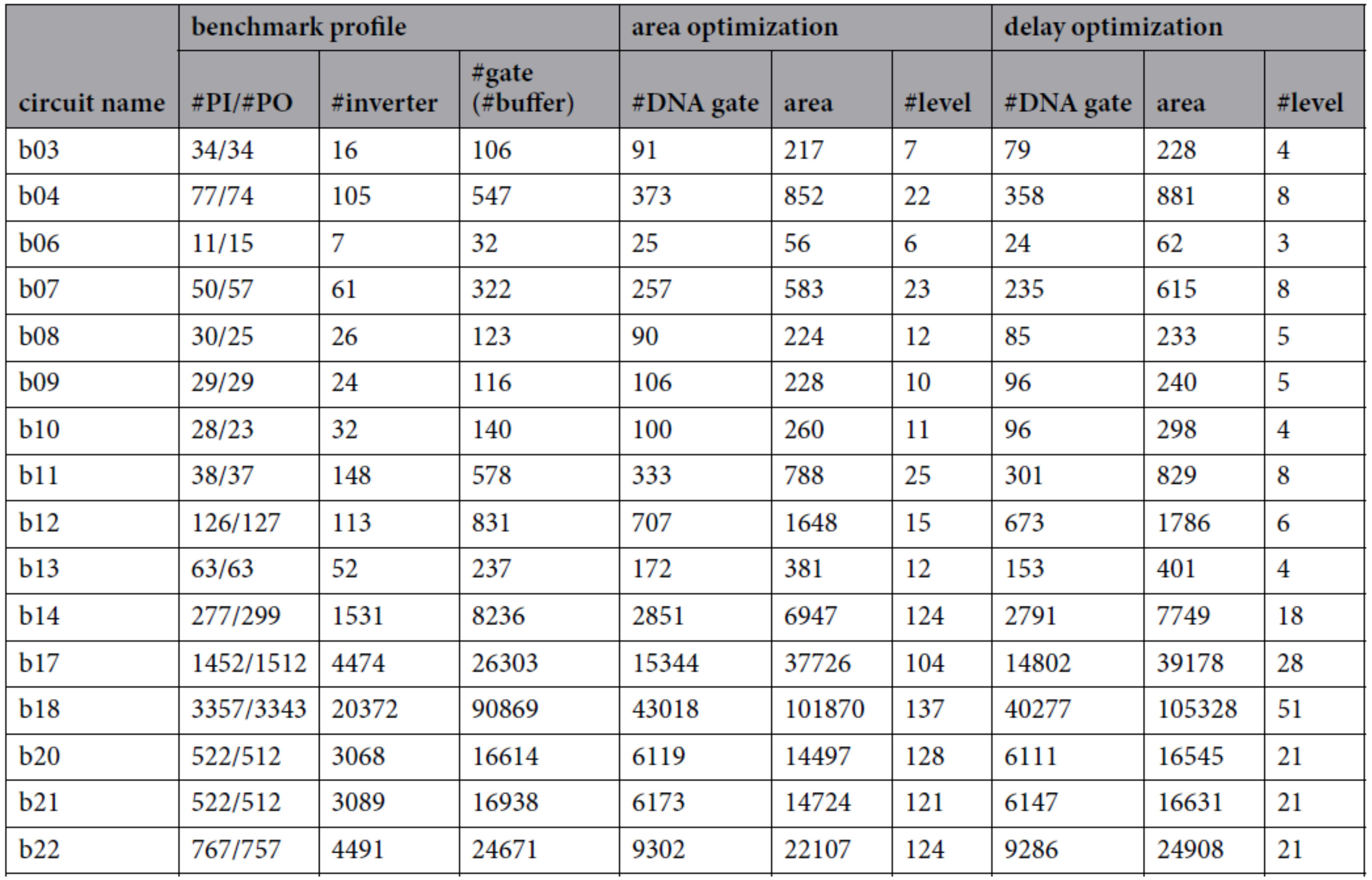


$$\begin{cases} A \to n7 \to Y, \\ B \to n7 \to Y, \\ B \to n8 \to Y, \\ B \to n8 \to Z, \\ C \to n8 \to Y, \\ C \to n8 \to Z. \end{cases}$$

$$\begin{cases} B \to n7 \to n8 \to Y, \\ C \to n7 \to n8 \to Y. \end{cases}$$

Synthesis Results

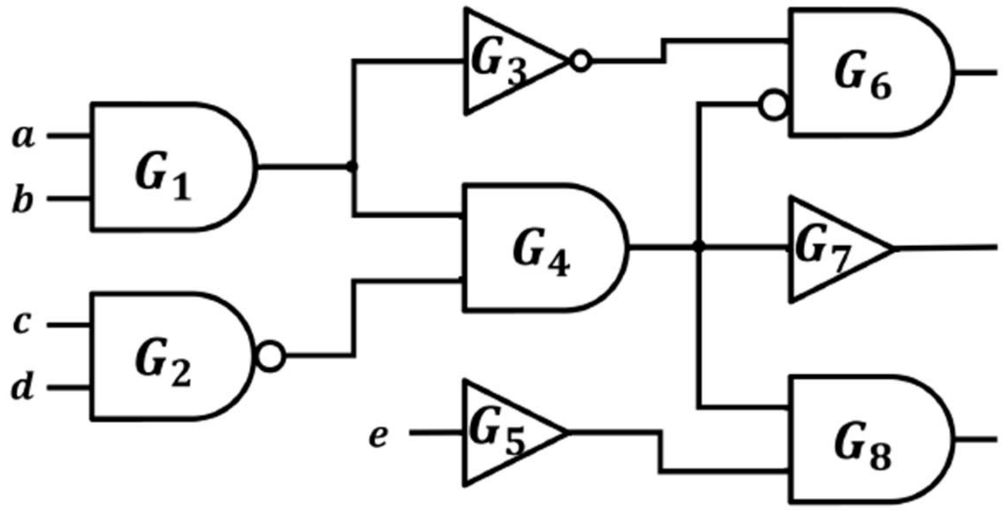
- Recombinase circuit synthesis with ABC



	benchmark profile			area optimization			delay optimization		
circuit name	#PI/#PO	#inverter	#gate (#buffer)	#DNA gate	area	#level	#DNA gate	area	#level
b03	34/34	16	106	91	217	7	79	228	4
b04	77/74	105	547	373	852	22	358	881	8
b06	11/15	7	32	25	56	6	24	62	3
b07	50/57	61	322	257	583	23	235	615	8
b08	30/25	26	123	90	224	12	85	233	5
b09	29/29	24	116	106	228	10	96	240	5
b10	28/23	32	140	100	260	11	96	298	4
b11	38/37	148	578	333	788	25	301	829	8
b12	126/127	113	831	707	1648	15	673	1786	6
b13	63/63	52	237	172	381	12	153	401	4
b14	277/299	1531	8236	2851	6947	124	2791	7749	18
b17	1452/1512	4474	26303	15344	37726	104	14802	39178	28
b18	3357/3343	20372	90869	43018	101870	137	40277	105328	51
b20	522/512	3068	16614	6119	14497	128	6111	16545	21
b21	522/512	3089	16938	6173	14724	121	6147	16631	21
b22	767/757	4491	24671	9302	22107	124	9286	24908	21

Outline

Blocking Terminators in Cascaded Circuits



$$\underbrace{d_a T_b G_1 T}_{G_1}\underbrace{d_c d_d G_2 T}_{G_2}\underbrace{P_{g_1} G_3 T}_{G_3}\underbrace{d_{g_1} T_{g_2} G_4 T}_{G_4}\underbrace{d_e G_5 T}_{G_5}$$

$$\underbrace{d_{g_3} L_{g_4} G_6 T}_{G_6}\underbrace{d_{g_4} G_7 T}_{G_7}\underbrace{d_{g_4} T_{g_5} G_8 T}_{G_8}$$

Optimization by Gate Merging

- Merging condition

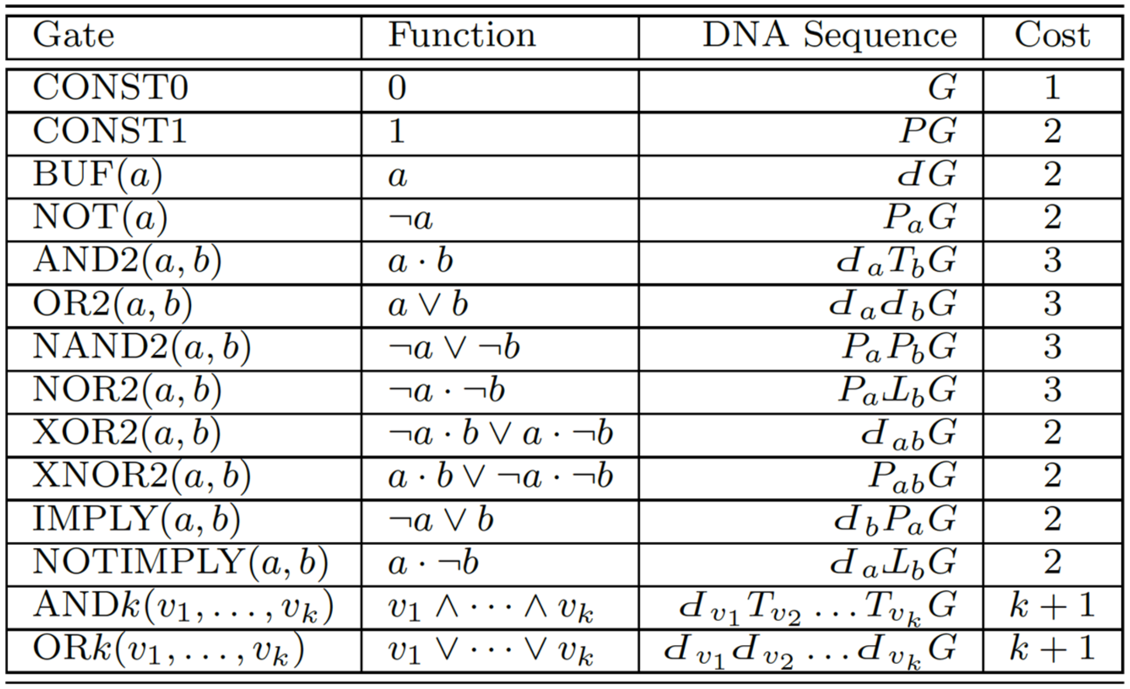
$$\underbrace{P_aP_bX\overbrace{T}^{\text{drop}}}_{X=\text{NAND2}(a,b)}\ \underbrace{\overbrace{d_x}^{\text{drop}}\ T_cYT}_{Y=\text{AND2}(x,c)}\quad \xrightarrow{\text{merge}}\quad P_aP_bXT_cYT$$

$$\xrightarrow{\text{if } |FO(X)|=1}\quad P_aP_bT_cYT$$

  - Gate Y starts with inversed promoter controlled by the output of gate X

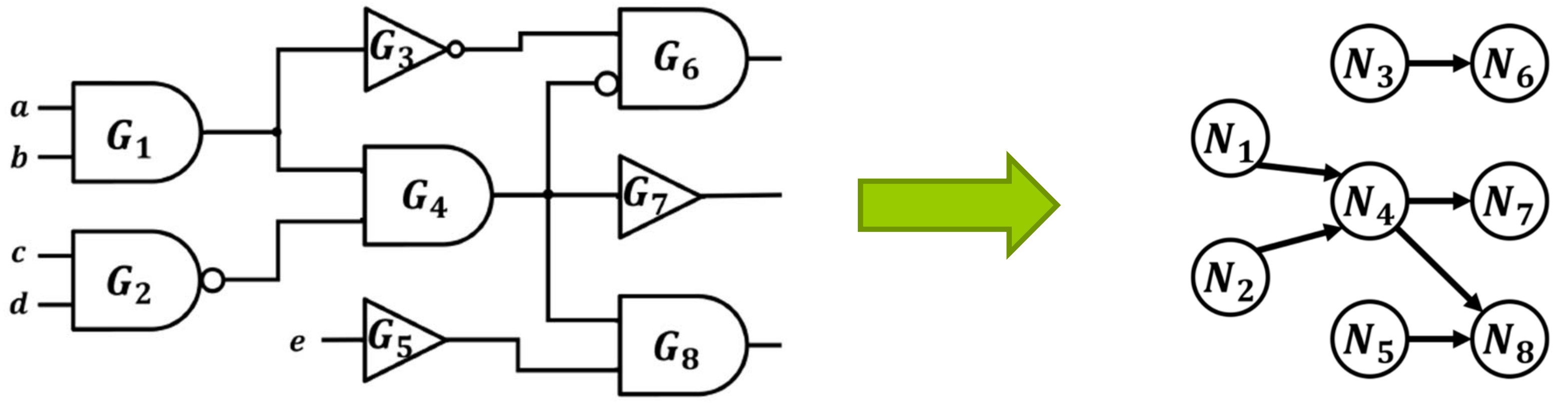
Gate-Merging Friendly Library

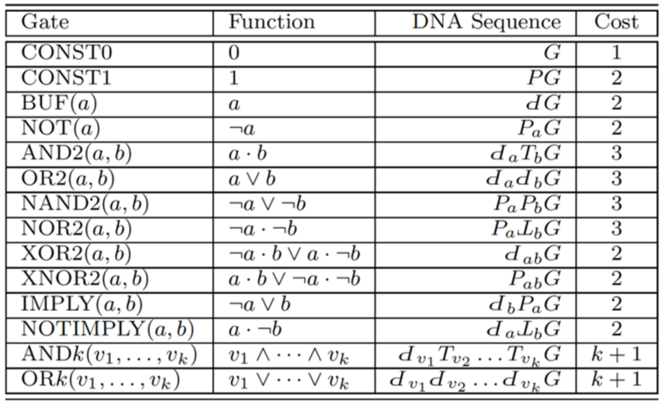
- Make DNA sequences start with inverted P



Gate	Function	DNA Sequence	Cost
CONST0	0	G	1
CONST1	1	PG	2
BUF(a)	a	dG	2
NOT(a)	$\neg a$	P_aG	2
AND2(a, b)	$a \cdot b$	d_aT_bG	3
OR2(a, b)	$a \vee b$	d_ad_bG	3
NAND2(a, b)	$\neg a \vee \neg b$	P_aP_bG	3
NOR2(a, b)	$\neg a \cdot \neg b$	$P_a\perp_bG$	3
XOR2(a, b)	$\neg a \cdot b \vee a \cdot \neg b$	$d_{ab}G$	2
XNOR2(a, b)	$a \cdot b \vee \neg a \cdot \neg b$	$P_{ab}G$	2
IMPLY(a, b)	$\neg a \vee b$	d_bP_aG	2
NOTIMPLY(a, b)	$a \cdot \neg b$	$d_a\perp_bG$	2
AND$k(v_1, \ldots, v_k)$	$v_1 \wedge \cdots \wedge v_k$	$d_{v_1}T_{v_2} \ldots T_{v_k}G$	$k+1$
OR$k(v_1, \ldots, v_k)$	$v_1 \vee \cdots \vee v_k$	$d_{v_1}d_{v_2} \ldots d_{v_k}G$	$k+1$

Mergeability Graph

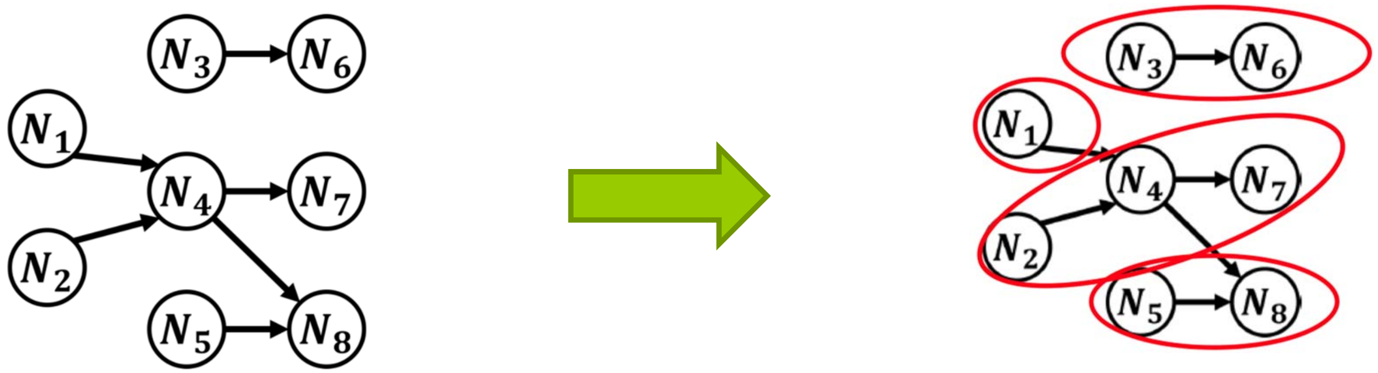




Gate	Function	DNA Sequence	Cost
CONST0	0	G	1
CONST1	1	PG	2
BUF(a)	a	dG	2
NOT(a)	$\neg a$	P_aG	2
AND2(a,b)	$a \cdot b$	d_aT_bG	3
OR2(a,b)	$a \vee b$	d_ad_bG	3
NAND2(a,b)	$\neg a \vee \neg b$	P_aP_bG	3
NOR2(a,b)	$\neg a \cdot \neg b$	P_aL_bG	3
XOR2(a,b)	$\neg a \cdot b \vee a \cdot \neg b$	$d_{ab}G$	2
XNOR2(a,b)	$a \cdot b \vee \neg a \cdot \neg b$	$P_{ab}G$	2
IMPLY(a,b)	$\neg a \vee b$	d_bP_aG	2
NOTIMPLY(a,b)	$a \cdot \neg b$	d_aL_bG	2
AND$k(v_1,\ldots,v_k)$	$v_1 \wedge \cdots \wedge v_k$	$d_{v_1}T_{v_2}\ldots T_{v_k}G$	$k+1$
OR$k(v_1,\ldots,v_k)$	$v_1 \vee \cdots \vee v_k$	$d_{v_1}d_{v_2}\ldots d_{v_k}G$	$k+1$

Path Covering on Mergeability Graph

- Weighted-path covering problem
  - Find a set of disjoint paths that covers all nodes
  - Can be transformed to the minimum assignment problem and solved by the Hungarian algorithm in $O(n^3)$



Path Covering by 0/1-ILP

$$
\begin{aligned}
\text{minimize} \quad & \sum_{g_i \in V} C(g_i) \\
\text{subject to} \quad & x_i + \sum_{g_j \in FO(g_i)} x_{i,j} = 1, \text{ for } i = 1, \ldots, n \\
& \sum_{g_i \in FI(g_j)} x_{i,j} \leq 1, \text{ for } j = 1, \ldots, n
\end{aligned}
$$

$$
C(g_i) = \begin{cases} 2 + |FI(g_i)|, & x_i = 1 \\ |FI(g_i)|, & x_i \neq 1 \wedge |FO(g_i)| \neq 1 \\ -1 + |FI(g_i)|, & x_i \neq 1 \wedge |FO(g_i)| = 1 \end{cases}
$$

$$
\begin{aligned}
\text{minimize} \quad & 2x_{1,4} + x_{2,4} + 2x_{4,7} + 2x_{4,8} + \\
& 4x_1 + 4x_2 + 3x_3 + 4x_4 + \\
& 4x_5 + 4x_6 + 3x_7 + 4x_8 \\
\text{subject to} \quad & x_1 + x_{1,4} = 1 \\
& x_2 + x_{2,4} = 1 \\
& x_3 + x_{3,6} = 1 \\
& x_4 + x_{4,7} + x_{4,8} = 1 \\
& x_5 + x_{5,8} = 1 \\
& x_6 = 1 \\
& x_7 = 1 \\
& x_8 = 1 \\
& x_{1,4} + x_{2,4} \leq 1 \\
& x_{4,8} + x_{5,8} \leq 1.
\end{aligned}
$$

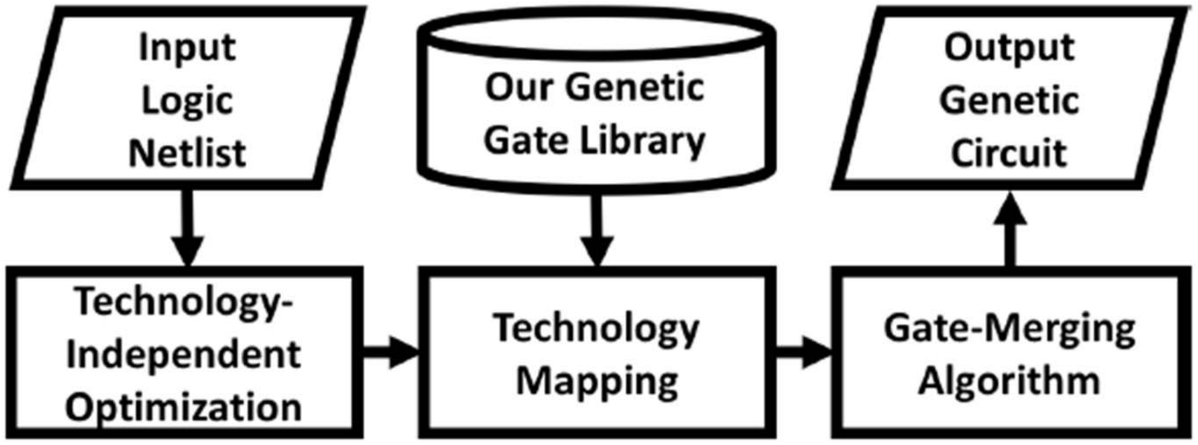
optimum solution $x_{2,4}, x_{4,7}, x_{5,8}, x_{3,6}, x_1, x_6, x_7, x_8 = 1$

$$
\underbrace{d_aT_bG_1T}_{G_1}\underbrace{d_cd_dG_2T}_{G_2}\underbrace{P_{g1}G_3T}_{G_3}\underbrace{d_{g1}T_{g2}G_4T}_{G_4}\underbrace{d_eG_5T}_{G_5}
$$

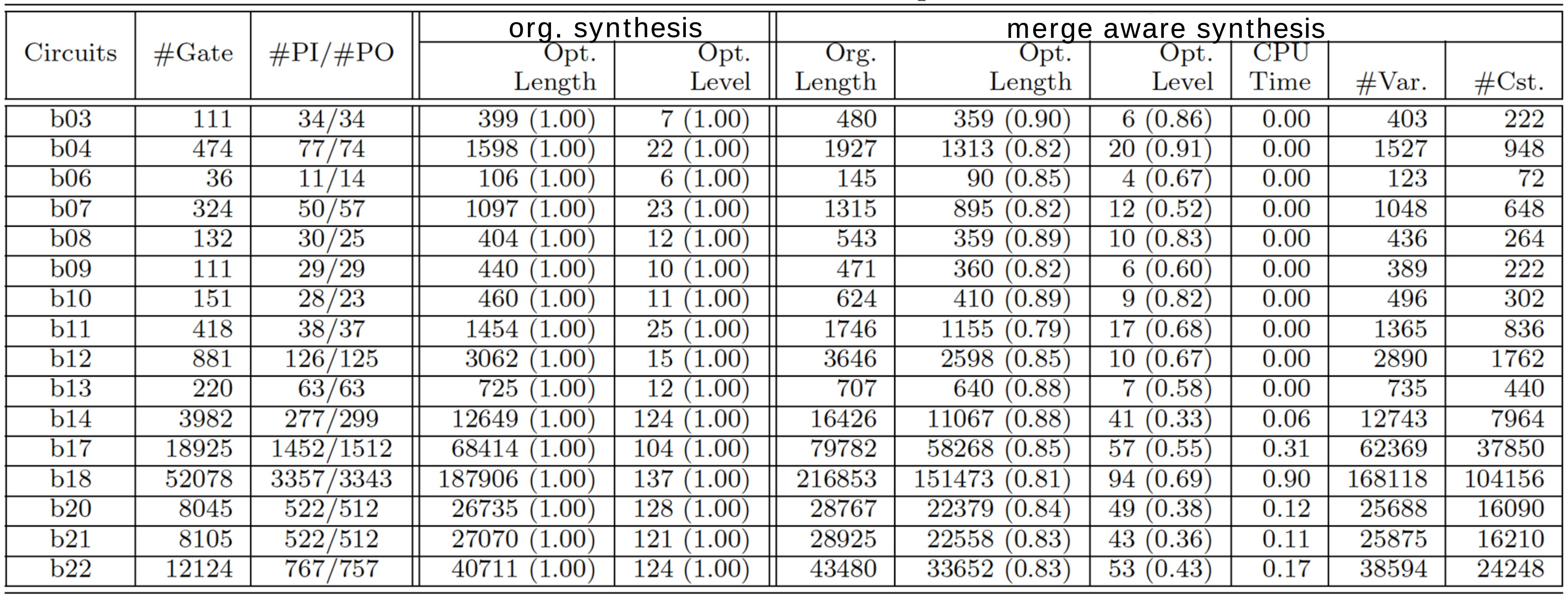
$$
\underbrace{d_{g3}L_{g4}G_6T}_{G_6}\underbrace{d_{g4}G_7T}_{G_7}\underbrace{d_{g4}T_{g5}G_8T}_{G_8}
$$

$$
\Longrightarrow \quad \underbrace{d_aT_bG_1T}_{G_1}\underbrace{P_cP_dT_{g1}G_4G_7T}_{G_2,G_4,G_7}\underbrace{P_{g1}L_{g4}G_6T}_{G_3,G_6}\underbrace{d_eT_{g4}G_8T}_{G_5,G_8}
$$

Optimization Flow



Optimization Results



Circuits	#Gate	#PI/#PO	org. synthesis		merge aware synthesis					
			Opt. Length	Opt. Level	Org. Length	Opt. Length	Opt. Level	CPU Time	#Var.	#Cst.
b03	111	34/34	399 (1.00)	7 (1.00)	480	359 (0.90)	6 (0.86)	0.00	403	222
b04	474	77/74	1598 (1.00)	22 (1.00)	1927	1313 (0.82)	20 (0.91)	0.00	1527	948
b06	36	11/14	106 (1.00)	6 (1.00)	145	90 (0.85)	4 (0.67)	0.00	123	72
b07	324	50/57	1097 (1.00)	23 (1.00)	1315	895 (0.82)	12 (0.52)	0.00	1048	648
b08	132	30/25	404 (1.00)	12 (1.00)	543	359 (0.89)	10 (0.83)	0.00	436	264
b09	111	29/29	440 (1.00)	10 (1.00)	471	360 (0.82)	6 (0.60)	0.00	389	222
b10	151	28/23	460 (1.00)	11 (1.00)	624	410 (0.89)	9 (0.82)	0.00	496	302
b11	418	38/37	1454 (1.00)	25 (1.00)	1746	1155 (0.79)	17 (0.68)	0.00	1365	836
b12	881	126/125	3062 (1.00)	15 (1.00)	3646	2598 (0.85)	10 (0.67)	0.00	2890	1762
b13	220	63/63	725 (1.00)	12 (1.00)	707	640 (0.88)	7 (0.58)	0.00	735	440
b14	3982	277/299	12649 (1.00)	124 (1.00)	16426	11067 (0.88)	41 (0.33)	0.06	12743	7964
b17	18925	1452/1512	68414 (1.00)	104 (1.00)	79782	58268 (0.85)	57 (0.55)	0.31	62369	37850
b18	52078	3357/3343	187906 (1.00)	137 (1.00)	216853	151473 (0.81)	94 (0.69)	0.90	168118	104156
b20	8045	522/512	26735 (1.00)	128 (1.00)	28767	22379 (0.84)	49 (0.38)	0.12	25688	16090
b21	8105	522/512	27070 (1.00)	121 (1.00)	28925	22558 (0.83)	43 (0.36)	0.11	25875	16210
b22	12124	767/757	40711 (1.00)	124 (1.00)	43480	33652 (0.83)	53 (0.43)	0.17	38594	24248

16% length reduction; 36% level reduction

Outline

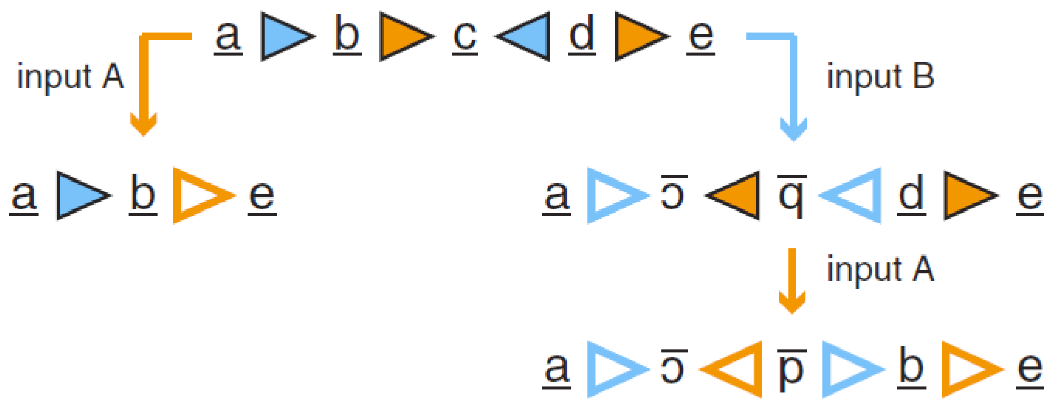
- Introduction
- Formalism
- Genetic Circuit Synthesis
- Genetic Circuit Optimization
- **Summary and future work**

Summary

- Motivated bio-design automation
- Formalized the syntax and semantics of recombinase-based logic gates
- Used existing logic synthesis flow for genetic circuit synthesis
- Optimized genetic circuit by gate merging

Future Work

- Recombinase-based genetic circuits
  - Nondeterministic and/or circuit synthesis



[Roquet et al. 2016]

- Analog computation
- Wet lab validation

References

- Tai-Yin Chu, J. Logic synthesis of recombinase-based genetic circuits. Scientific Reports (to appear).
- Chun-Ling Lai, J., Francois Fages. Recombinase-based genetic circuit optimization. BioCAS, 2017.

Acknowledgement

- Collaborators
  - Tai-Yin Chiu, NTU
  - Chun-Ning Lai, NTU
  - Francois Fages, INRIA
  - Franck Molina, CNRS

- Sponsor
  - Ministry of Science of Technology, Taiwan

Thanks for Your Attention!

- Questions?