Supplementary Information

Temporal logic circuits implementation using dual cross-inhibition mechanism based on DNA strand displacement

Yuan Liu, [a] Xiaokang Zhang, [a] Xun Zhang, [a] Xin Liu, [a] Bin Wang, [b] Qiang Zhang *[a] and Xiaopeng Wei *[a]

a School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China.

b Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian 116622, China.

Contents

S1. Kinetic characterization of the temporal OR gate

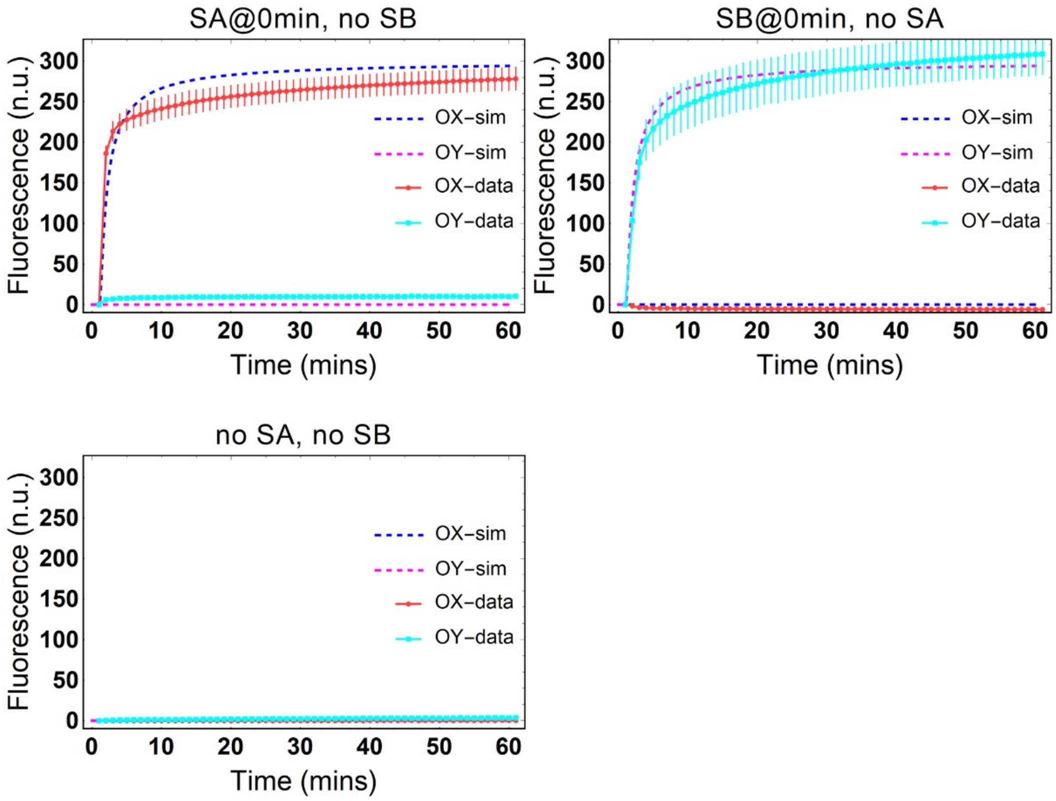

Fig. S1. Kinetic characterization of the temporal OR gate. The results of real-time fluorescence experiments, accompanied by simulated results for three cases including SA input only, SB input only and no input. Detector A, Detector B, Reporter X and Reporter Y are added as substrates in advance. [DA] = [DB] = [RX] = [RY] = 300nM, [SA] = [SB] = 600nM. Fluorescence values in the experimental data were collected every minute.

S2. Kinetics characterization of temporal OR gate at time intervals of 10 mins and 20 mins

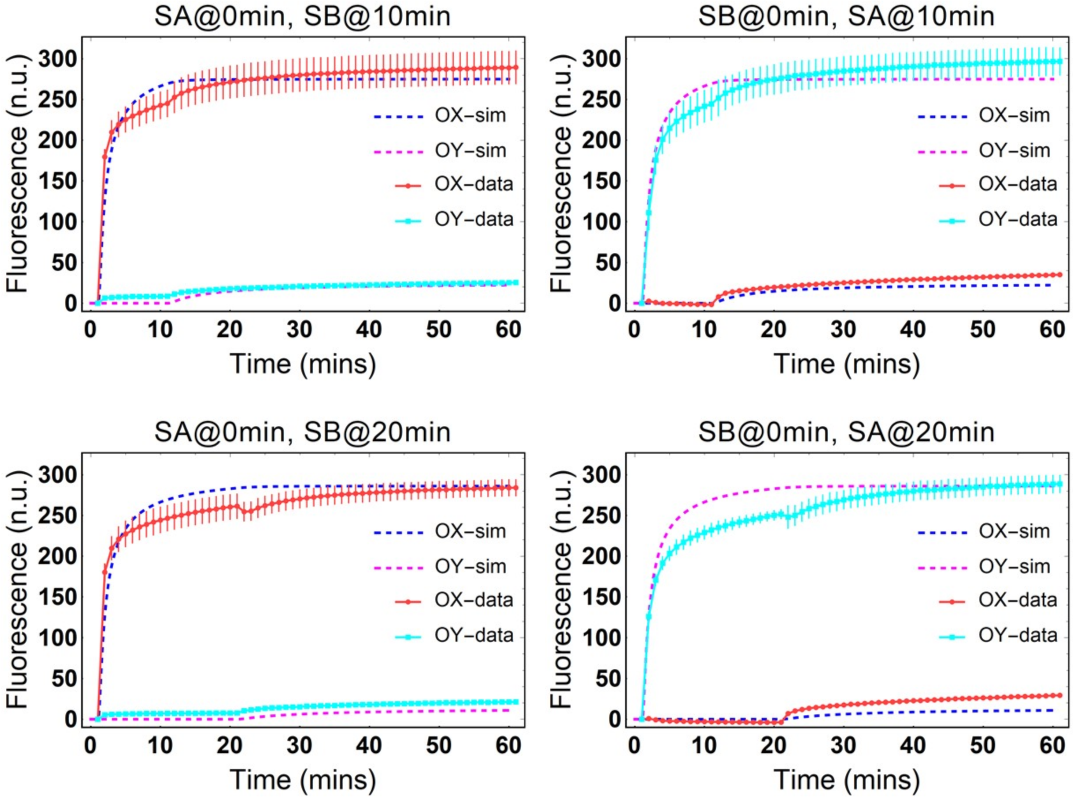

Fig. S2. Real-time fluorescence and simulation results with time intervals of 10 minutes and 20 minutes between SA and SB. The order of SA and SB additions, as well as the time intervals, are indicated above each plot. The terms "OX-data" and "OY-data" represent the experimental data for OX and OY, while "OX-sim" and "OY-sim" represent the simulation results for OX and OY. In all experiments and simulations, [DA] = [DB] = [RX] = [RY] = 300nM, [SA] = [SB] = 600nM. DA, DB, RX and RY are added as substrates in advance.

S3. The effect of reaction rate constants k_1 and k_{-1}

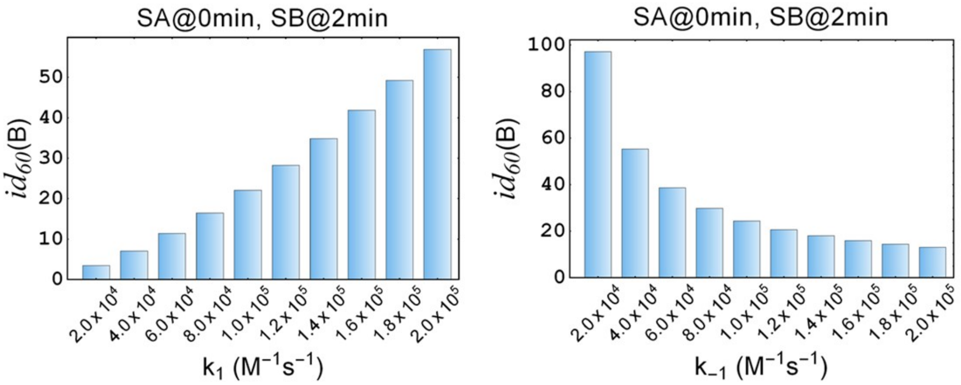

Fig. S3. The effect of reaction rate constants k_1 and k_{-1} on the inhibition degree id_{60}(B) of temporal OR gate with incorporation of the annihilation gate. X-axis represents the reaction rate constant k_1 and k_{-1}, while the Y-axis represents the inhibition degree. Here, $k_2 = 1.7\times10^5$ $M^{-1}s^{-1}$, $k_3 = 4\times10^5$ $M^{-1}s^{-1}$, $k_a = 10^{13}$ $M^{-2}s^{-1}$, [DA] = [DB] = [RX] = [RY] = [AG] = 300nM, [SA] = [SB] = 600nM; for the left figure, k_{-1}=1.7×10^5 $M^{-1}s^{-1}$, and for the right figure, k1=7.5×10^4 $M^{-1}s^{-1}$.

S4. Slice plots of the surface plot

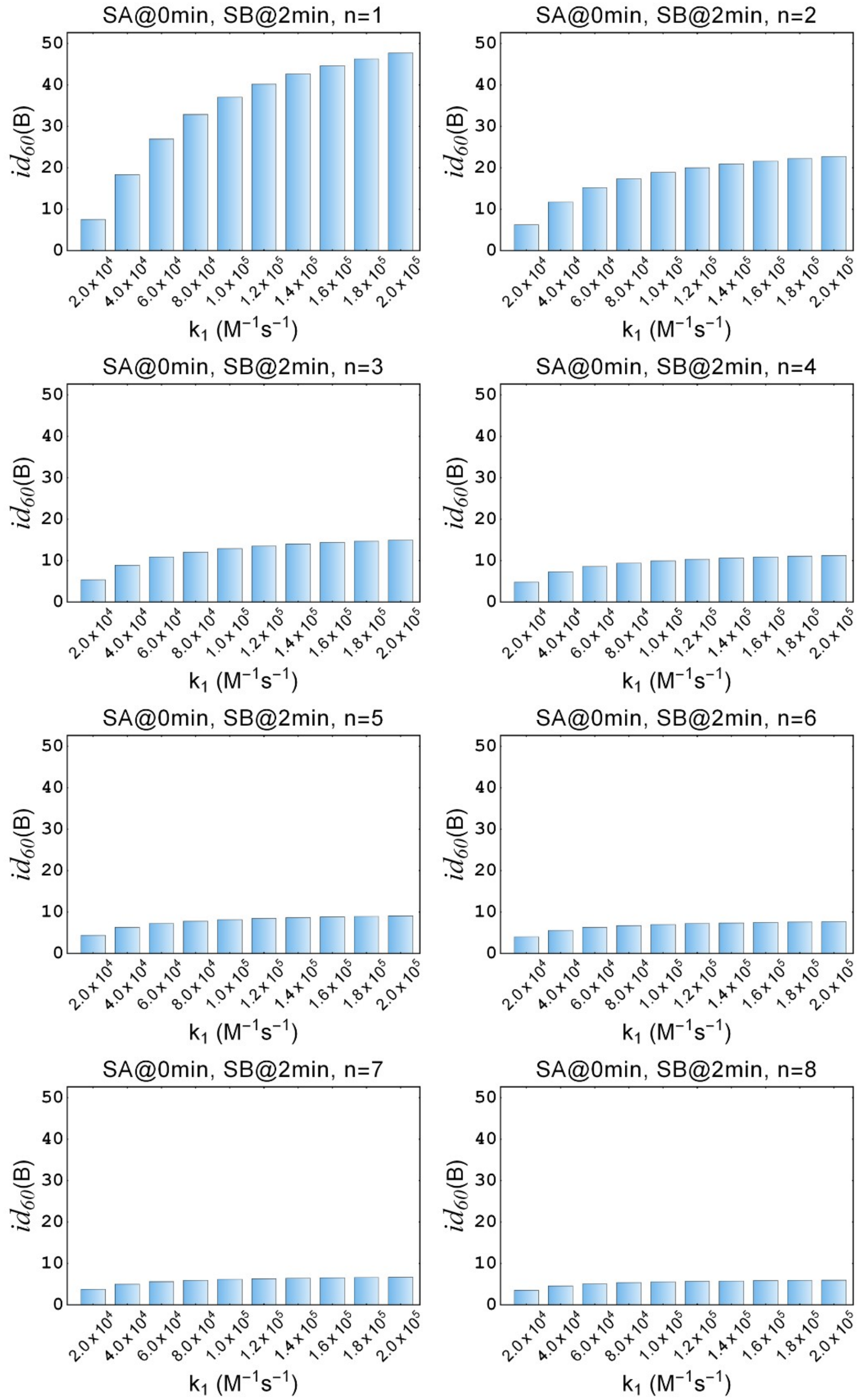

Fig. S4. Slice plots of the surface plot depicting the relationship between k1, k2, and the degree of inhibition. In the plot label, n indicates that k_2 is n times as large as k_1. The value of n is labeled in the title of each subfigure. In all subfigures, SA is added 2 minutes before SB. $k_2 = 1.7\times10^5$ $M^{-1}s^{-1}$, $k_3 = 4\times10^5$ $M^{-1}s^{-1}$, $k_a = 10^{13}$ $M^{-2}s^{-1}$, [DA] = [DB] = [RX] = [RY] = 300nM, [SA] = [SB] = 600nM.

S5. The effect of global concentration variation on the inhibition degree

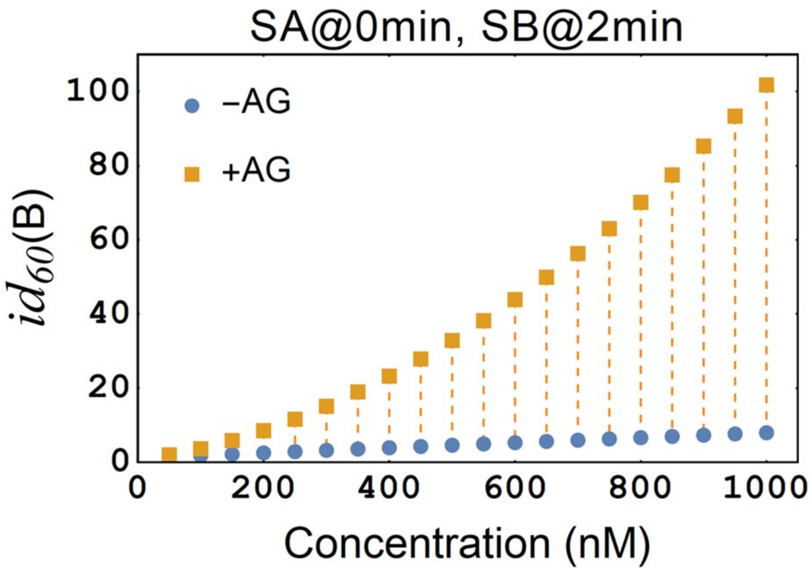

Fig. S5. The effect of global concentration variation on the inhibition degree. X-axis represents the concentrations of the substrates and the annihilation gate. Y-axis represents the inhibition degree. The ratio of the input to the substrate is kept at a constant of 2. The concentration of the annihilation gate is equal to the concentration of the four substrates. [DA] = [DB] = [RX] = [RY] = [AG], k1=7.5×10^4 M^{-1}s^{-1}, k$_{-1}$=1.7×10^5 M^{-1}s^{-1}, k_2 = 1.7×10^5 M^{-1}s^{-1}, k_3 = 4×10^5 M^{-1}s^{-1}, k_a = 10^{13} M^{-2}s^{-1}.

S6. The effect of the variation in the ratio of input to substrate

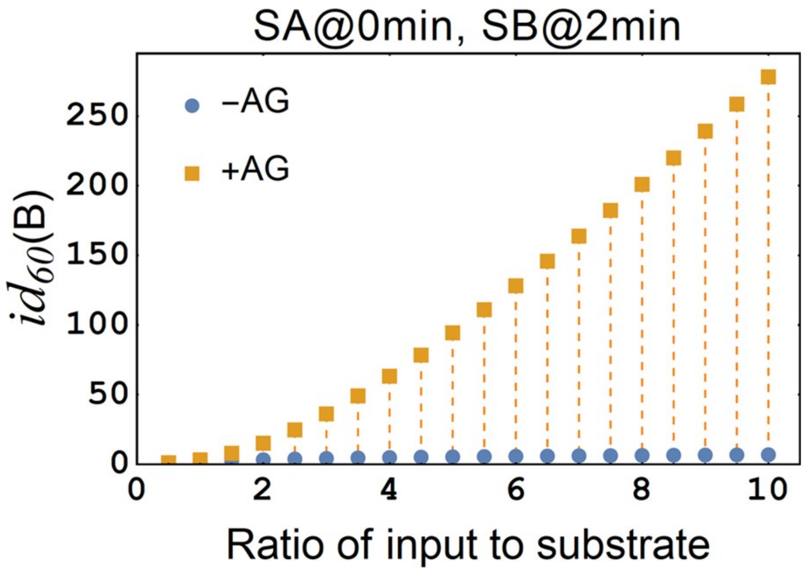

Fig. S6. The effect of the variation in the ratio of input to substrate on the degree of inhibition with the concentration of substrates being constant. The concentration of the annihilation gate is equal to the concentration of the four substrates (DA, DB, RX and RY). [DA] = [DB] = [RX] = [RY] = [AG], k1=7.5×10^4 M^{-1}s^{-1}, k$_{-1}$=1.7×10^5 M^{-1}s^{-1}, k$_2$ = 1.7×10^5 M^{-1}s^{-1}, k$_3$ = 4×10^5 M^{-1}s^{-1}, k$_a$ = 10^{13} M^{-2}s^{-1}.

S7. Investigation of the relationship between substrate concentration and inhibition degree

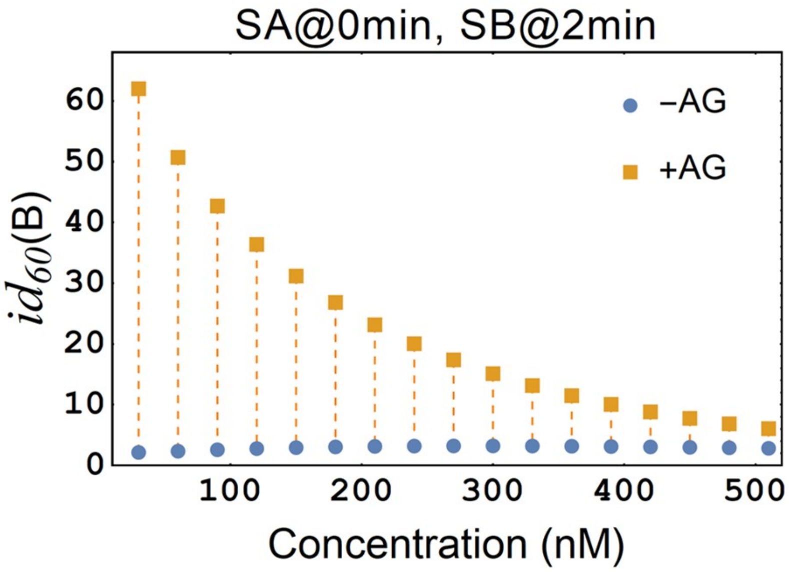

Fig. S7. Investigation of the relationship between substrate concentration and inhibition degree under fixed input and AG concentration. The concentrations of the input and the annihilation gate are fixed as constants. X-axis represents the concentration of the substrates DA, DB, RX and RY, while the Y-axis represents the inhibition degree. [DA] = [DB] = [RX] = [RY], [AG] = 300nM, [SA] = [SB] = 600nM, k1=7.5×10^4 $M^{-1}s^{-1}$, k_{-1}=1.7×10^5 $M^{-1}s^{-1}$, k_2 = 1.7×10^5 $M^{-1}s^{-1}$, k_3 = 4×10^5 $M^{-1}s^{-1}$, k_a = 10^{13} $M^{-2}s^{-1}$.

S8. DNA sequences

We used the analysis module of NUPACK (www.nupack.org) to analyze the DNA sequences used, in order to minimize potential crosstalk and leakage.

Table S1. DNA sequences and modifications

DNA strand	Sequence (5’ to 3’)	Domains	Length (nt)
SA	TACATTCA**TCGTAC**CATCCACTAGCACACACTGAGAC	c1, t, b	37
DA5 (SY)	TCTCAAATCCTCTGTAACGTTGGACTCTCGTCACTG	y, a	36
DA3 (KA)	CATCCACTAGCACACACTGAGAC**ATGCAC**	b, s	29
DAD	**GTGCAT**GTCTCAGTGTGTGCTAGTGGATG**GTACGA**CAGTGACGAGAGTCCAACGTTAC	a*, t*, b*, s*	58
SB	TAGTATCT**ATGCAC**GTAACGTTGGACTCTCGTCACTG	c2, s, a	37
DB5 (SX)	TTCTACTCTACCACATCCACTAGCACACACTGAGAC	x, b	36
DB3 (KB)	GTAACGTTGGACTCTCGTCACTG**TCGTAC**	a, t	29
DBD	**GTACGA**CAGTGACGAGAGTCCAACGTTAC**GTGCAT**GTCTCAGTGTGTGCTAGTGGATG	b*, s*, a*, t*	58
XRU (OX)	(FAM)-TTCTACTCTACCACATCCA	x, b1	19
XRD	**TGCTAG**TGGATGTGGTAGAGTAGAA-(BHQ1)	x*, b2*	25
YRU (OY)	(HEX)-TCTCAAATCCTCTGTAACG	y, a1	19
YRD	**GTCCAA**CGTTACAGAGGATTTGAGA-(BHQ1)	y*, a2*	25

Notes: The sequences of the toeholds are highlighted in bold.

Table S2. Composition of all complexes

Complex	DNA strands
DA (Detector A)	DA5, DA3, DAD
DB (Detector B)	DB5, DB3, DBD
RX (Reporter X)	XRU, XRD
RY (Reporter Y)	YRU, YRD

S9. The Mathematica modelling codes used for the simulation

The simulation codes were running on Wolfram Mathematica 12.0. (Link for CRNSimulator: https://users.ece.utexas.edu/~soloveichik/crnsimulator.html.

Modelling codes for temporal OR gate:

```
Note: requires CRNSimulator.m and CRNSimulatorExtensions.m to be loaded.
<< "D:\\dir\\CRNSimulator.m"
<< "D:\\dir\\CRNSimulatorExtensions.m"
Clear[prinArray];
prinMatrix = Import["D:\\dir\\figure1.xlsx"];
nLength = 61;
prinArray=Table[0*i*j,{i,24},{j,nLength}];
For[j=1,j<nLength+1,j++,
  For[i=1,i<24+1,i++,prinArray[[i,j]]=prinMatrix[[1,j,i]]/9.8]
  ];
curve1=Table[Around[prinArray[[1,n]],prinArray[[5,n]]],{n,61}];
curve2=Table[Around[prinArray[[2,n]],prinArray[[6,n]]],{n,61}];
curve3=Table[Around[prinArray[[3,n]],prinArray[[7,n]]],{n,61}];
curve4=Table[Around[prinArray[[4,n]],prinArray[[8,n]]],{n,61}];
k1=7.5*10^4;
k2=1.7*10^5;
k3=1.7*10^5;
k4=4*10^5;
nSub=300;
nInput=600;
rsys={
  revrxn[SA+DA,W1+KA,k1,k2],
  revrxn[SB+DB,W2+KB,k1,k2],
  rxn[KA+DB,SX+W4,k3],
  rxn[KB+DA,SY+W3,k3],
  rxn[SX+RX,OX+W5,k4],
  rxn[SY+RY,OY+W6,k4],
  conc[DA,nSub*10^-9],
  conc[DB,nSub*10^-9],
  conc[RX,nSub*10^-9],
  conc[RY,nSub*10^-9]
  };
schedule1={
  {1*60,0,0},
  {30*60,nInput*10^-9,0},
  {30*60,0,nInput*10^-9}
  };
tmax=60*60;
```

```
sol=SimulateRxnsysWithSchedule[rsys,schedule1,{SA,SB}];
plotter1={OX[t*60]*10^9,OY[t*60]*10^9}/.sol;
xoSimu1 = Plot[plotter1,{t,0,tmax/60},PlotRange->{0,All},
  PlotStyle->{{Thick,RGBColor["#0000FF"],Dashed},{Thick,RGBColor["#FF02FF"],Dashed}}];
xoExpe1 = ListLinePlot[{curve1,curve2},
  PlotStyle->{RGBColor["#FF4040"],Cyan},
  PlotMarkers->{Automatic, 4.5}];
```

Modelling codes for time-response characteristics:

```
figTime1={};
figTime2={};
data=Table[
  k1=7.5*10^4;
  k2=1.7*10^5;
  k3=1.7*10^5;
  k4=4*10^5;
  nSub=300;
  nInput=600;
  rsys={
    revrxn[SA+DA,W1+KA,k1,k2],
    revrxn[SB+DB,W2+KB,k1,k2],
    rxn[KA+DB,SX+W4,k3],
    rxn[KB+DA,SY+W3,k3],
    rxn[SX+RX,OX+W5,k4],
    rxn[SY+RY,OY+W6,k4],
    conc[DA,nSub*10^-9],
    conc[DB,nSub*10^-9],
    conc[RX,nSub*10^-9],
    conc[RY,nSub*10^-9]};
  schedule={
    {0*60,0,0},
    {x*60,nInput*10^-9,0},
    {60*60,0,nInput*10^-9}};
  tmax=60*60;
  sol=SimulateRxnsysWithSchedule[rsys,schedule,{SA,SB}];
  AppendTo[figTime1 ,{x,OX[tmax]/OY[tmax]/.sol}];
  AppendTo[figTime2,{x,W1[tmax]/W2[tmax]/.sol}],
  {x,10,50,1}];
```

Modelling codes for the temporal OR gate with the incorporation of Annihilation Gate:

```
ClearAll;
k1=7.5*10^4;
k2=1.7*10^5;
k3=1.7*10^5;
```

```
k4=4*10^5;
nSub=150;
nInput=300;
nAnni =150;
intervalTime=2;
rsys1={
  revrxn[SA+DA,W1+KA,k1,k2],
  revrxn[SB+DB,W2+KB,k1,k2],
  rxn[KA+DB,SX+W4,k3],
  rxn[KB+DA,SY+W3,k3],
  rxn[SX+RX,OX+W5,k4],
  rxn[SY+RY,OY+W6,k4],
  conc[DA,nSub*10^-9],
  conc[DB,nSub*10^-9],
  conc[RX,nSub*10^-9],
  conc[RY,nSub*10^-9]};
schedule={
  {0*60,0,0},
  {intervalTime*60,nInput*10^-9,0},
  {60*60,0,nInput*10^-9}};
tmax=60*60;
sol=SimulateRxnsysWithSchedule[rsys1,schedule,{SA,SB}];
plotter1={OX[t*60]*10^9,OY[t*60]*10^9}/.sol;
timeRes1={OX[tmax]/OY[tmax]}/.sol;
cciSim=Plot[plotter1,{t,0,tmax/60},PlotRange->{0,All},
  PlotStyle->{{Thick,RGBColor["#29ABE2"]},{Thick,RGBColor["#FF00FF"]}},
  PlotLegends->Placed[{Text[Style["OX-sim",16]],Text[Style["OY-sim",16]]},{0.81,0.65}]];
ka=10^13;
rsys2={
  revrxn[SA+DA,W1+KA,k1,k2],
  revrxn[SB+DB,W2+KB,k1,k2],
  rxn[KA+KB+ANNI,W7+W8,ka],
  rxn[KA+DB,SX+W4,k3],
  rxn[KB+DA,SY+W3,k3],
  rxn[SX+RX,OX+W5,k4],
  rxn[SY+RY,OY+W6,k4],
  conc[DA,nSub*10^-9],
  conc[DB,nSub*10^-9],
  conc[RX,nSub*10^-9],
  conc[RY,nSub*10^-9],
  conc[ANNI,nAnni*10^-9]};
schedule={
  {0*60,0,0},
  {intervalTime*60,nInput*10^-9,0},
  {60*60,0,nInput*10^-9}};
tmax=60*60;
sol=SimulateRxnsysWithSchedule[rsys2,schedule,{SA,SB}];
plotter2={OX[t*60]*10^9,OY[t*60]*10^9}/.sol;
timeRes2={OX[tmax]/OY[tmax]}/.sol;
cciSimAnni=Plot[plotter2,{t,0,tmax/60},PlotRange->{0,All},
  PlotStyle->{{Thick,RGBColor["#29ABE2"]},{Thick,RGBColor["#FF00FF"]}},
  PlotLegends->Placed[{Text[Style["OX-sim",16]],Text[Style["OY-sim",16]]},{0.81,0.65}] ];
```