

Logisim(로지심)

소개 및

간단한 사용법

※주 : 본 내용은 로지심 프로그램 내부 Help 메뉴에 있는 내용을 중심으로 번역, 정리 한 내용이며 일부 내용은 송상훈 저 '논리회로 설계' (생능출판사)를 참조하여 정리된 내용입니다. – 오세춘 –

참고 동영상 :

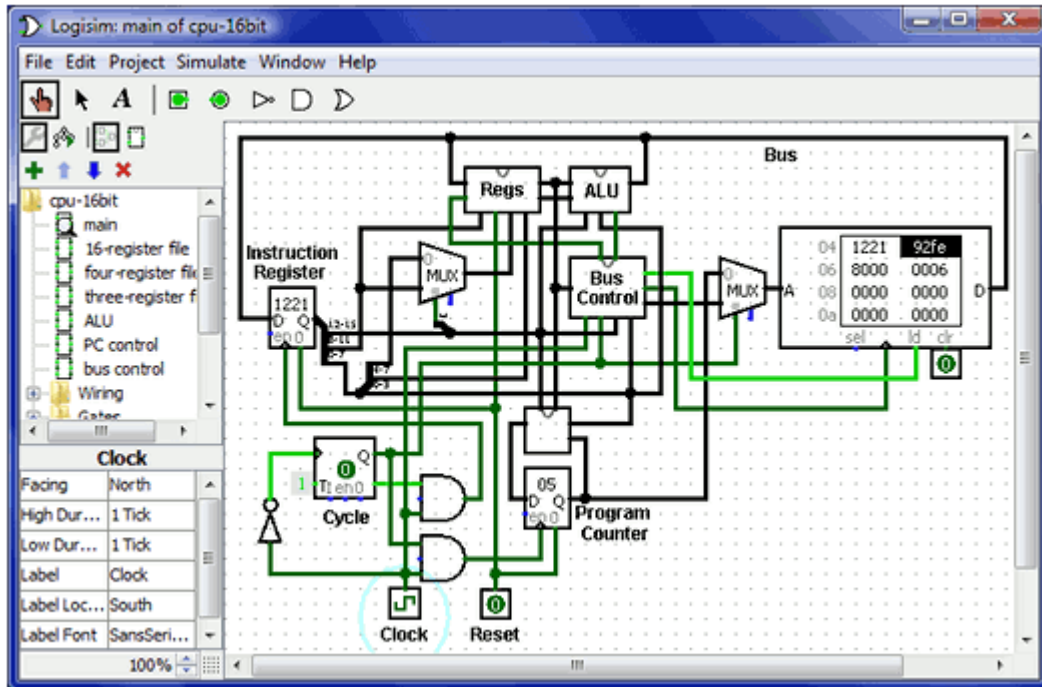
https://www.youtube.com/watch?v=cMz7wyY_PxE

<https://www.youtube.com/watch?v=KS5UhAUzhT8>

<https://www.youtube.com/watch?v=YCBa1NH4ORE>

Logisim 은 디지털 회로의 설계 및 시뮬레이션이 가능한 교육용 툴이며 쉬운 사용자 인터페이스는 물론 내장된 회로를 이용해 쉽게 사용할 수 있다.

작은 규모의 회로는 물론 큰 규모(교육용 CPU)의 회로까지 회로를 그리고 시뮬레이션 할 수 있으며 마우스로 간단히 선들을 이어줄 수 있다



주요 기능

- 무료 소프트웨어 (Logisim is open-source (GPL).)
- Java 5 이상이 설치 되어 있으면 실행 가능 (MacOS X, Windows)
- 그리기 기능은 직감적인 툴바(toolbar)를 기본으로 구성되어 있으며 배선 색깔의 변화는 시뮬레이션이나 디버깅 시 도움이 된다.
- 배선 tool(wiring tool)은 수평, 수직방향으로 자동으로 부품을 연결해 주기 때문에 매우 사용이 쉽니다.
- 완성한 회로는 파일 또는 GIF 형태의 파일로 저장이 되며 프린터로의 출력도 가능하다
- 완성된 회로는 다른 큰 회로를 그릴 때 하나의 부품처럼 부회로 (subcircuits) 형태로 활용이 가능하다. →계층적 구조(hierarchical)
- 기본적인 회로 부품, 입출력 장치, 게이트, multiplexers, arithmetic circuits, flip-flops, and RAM memory 등을 프로그램 내부에 포함하고 있다
- 내부 기능으로 제공되는 "combinational analysis" 모듈을 사용하면 회로도 와 진리표 그리고 부울 방정식 사이의 변환이 가능하다

1. 초심자용 사용법

Welcome to Logisim!

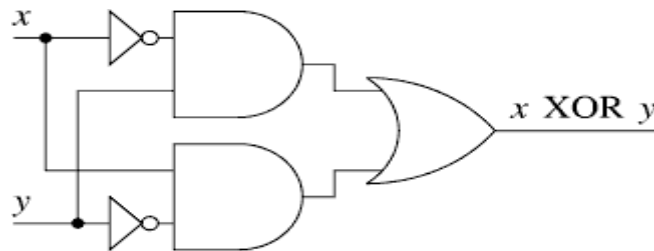
Logisim 은 디지털 회로의 설계 및 시뮬레이션 툴이며 디지털 회로가 어떻게 동작하는 지를 배울 수 있는 좋은 교육용 툴이다.

로직심을 연습하기 위해 XOR 회로를 구성해 보자

XOR 회로는 2 개의 입력을 비교해 그 값이 다를 경우에만 1 을 출력하는 회로이다, 다음의 진리표(truth Table) 참조

x	y	$x \text{ XOR } y$
0	0	0
0	1	1
1	0	1
1	1	0

아마 여러분은 다음과 같은 회로를 종이에 그리게 된다..



그러나 종이 위에 그릴 경우 검증할 수 없기 때문에 로직심을 이용해 회로를 그리면 테스트를 할 수 있을 뿐만 아니라 훨씬 훌륭히 그릴 수 있다.

Step 0: Orienting yourself (직접 회로를 그려보기)

Step 1: Adding gates (게이트 추가하기)

Step 2: Adding wires (배선을 연결하기)

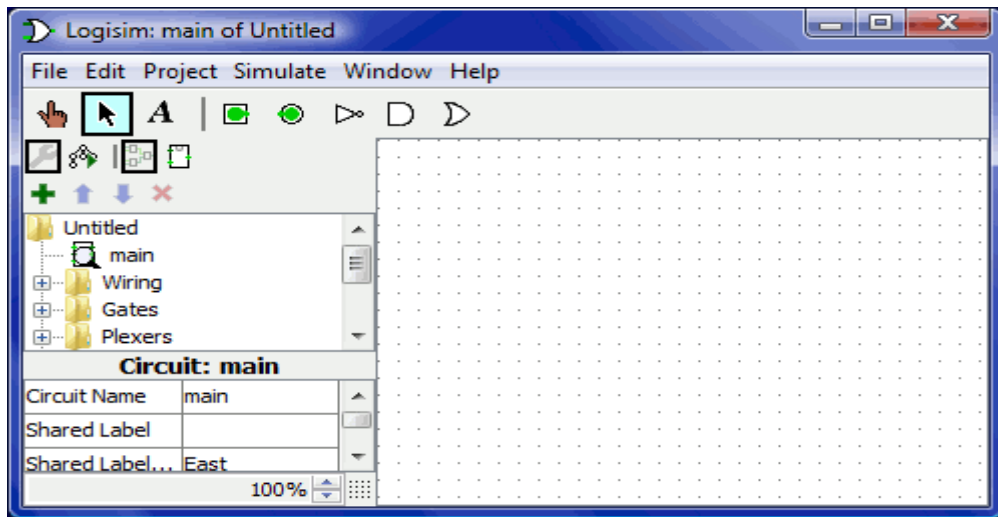
Step 3: Adding text (문자를 추가 하기)

Step 4: Testing your circuit (회로를 테스트 하기)

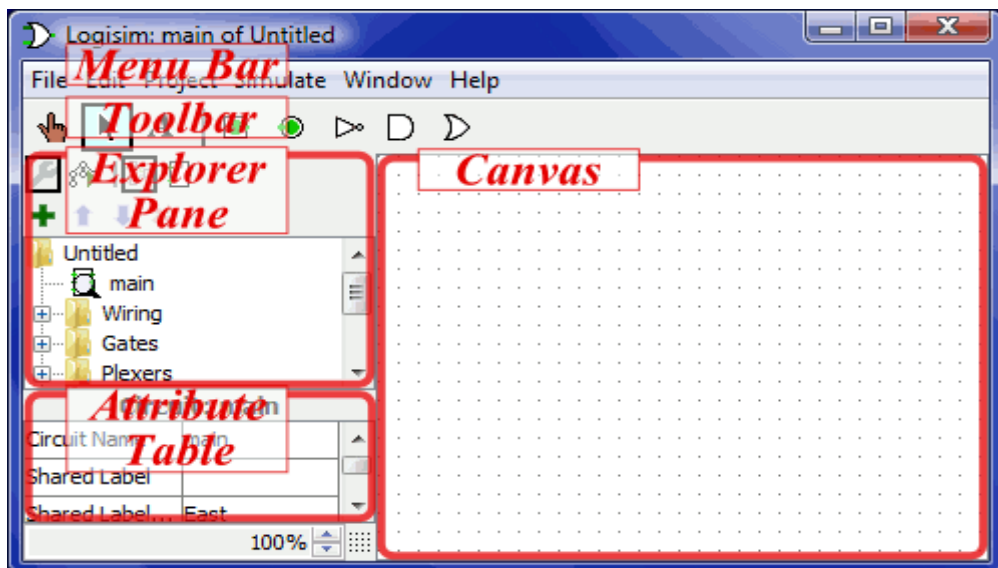
Enjoy your circuit-building!

Step 0: Orienting yourself (직접 회로를 그려보기)

로직심을 실행 시키면 아래와 같은 화면이 나온다. 시스템에 따라 약간 다르게 나올 경우도 있다.



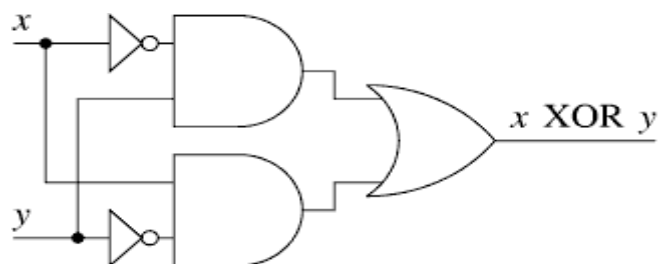
로지심의 화면은 크게 3 개의 파트로 나누어지는데 각 파트들은 *explorer pane*, *attribute table* 그리고 *canvas* 라 부른다. 이 각각의 파트 위에 메뉴바와 툴바가 표시된다.



여러분은 툴바를 이용해 canvas 파트에 회로를 그린다.

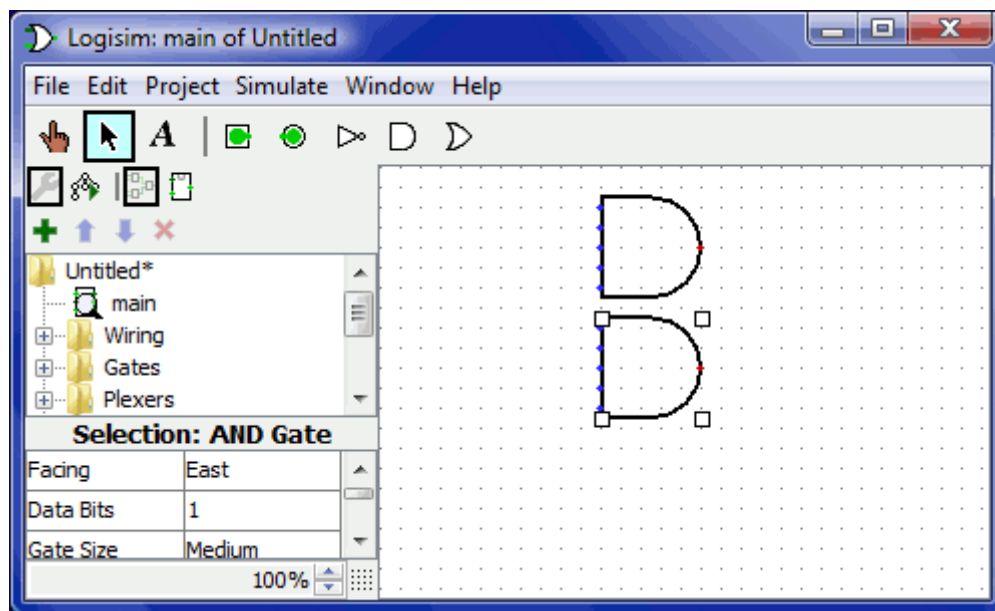
Step 1: Adding gates (게이트 추가하기)

로지심을 이용해 앞에서 다루었던 다음과 같은 XOR 회로를 그려보자



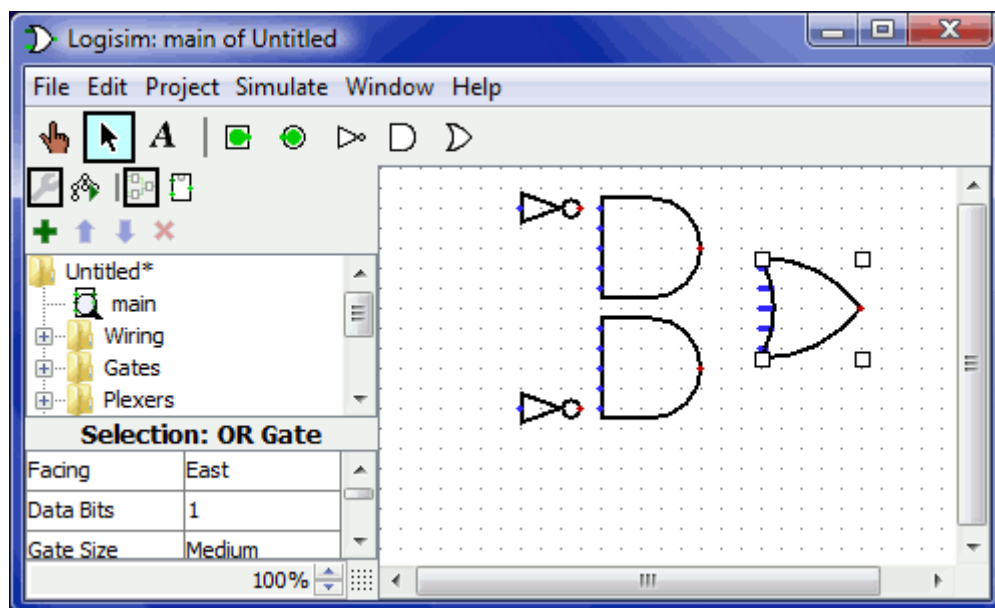
가장 처음으로 게이트를 먼저 그리고 그 다음에 배선을 하는 것이 낫다.

제일 처음으로 할 일은 2 개의 AND 게이트를 추가하는 것인데 툴바에 있는 AND 툴 그림(□, 툴 목록의 뒤에서 두번째 그림)을 누르고 다시 canvas 위로 마우스를 움직여 배치를 원하는 곳에서 클릭하면 AND 게이트가 추가 된다. 이때 왼쪽 부분에 일정부분의 공간을 두는 것이 필요하다. 그리고 지금 한 방법처럼 또 하나의 AND 게이트를 배치한다.



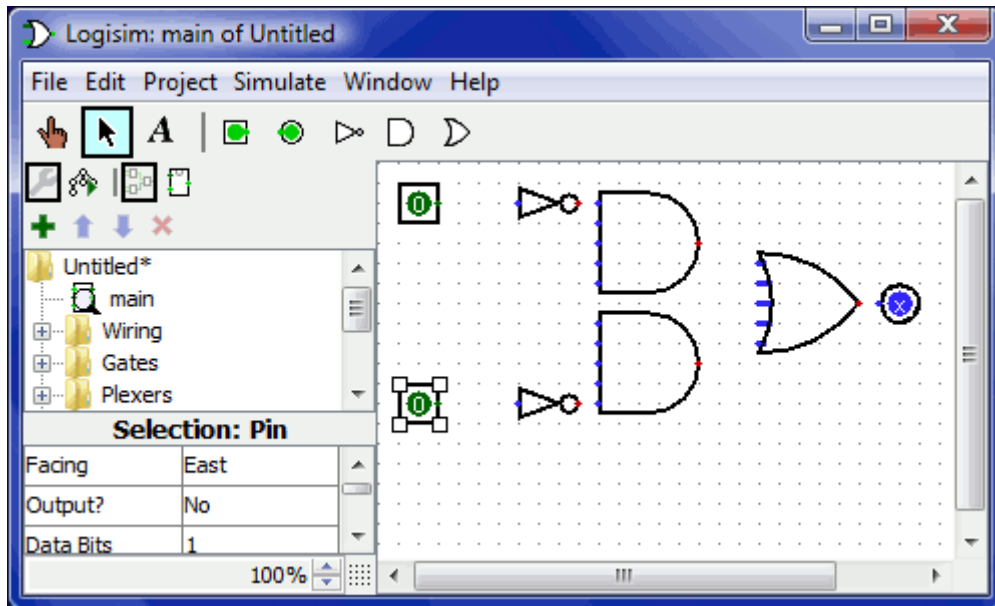
AND 게이트의 왼쪽에 5 개의 점이 보이는데 이 부분에 배선을 연결하게 된다. 지금 우리는 단지 2 개의 입력만을 사용하지만 향후 여러분의 용도에 따라 여러 개의 입력을 사용할 수 있다.

이제 다른 게이트도 추가해 보자. OR tool(▷)을 클릭하고 배치를 원하는 곳에 가서 다시 클릭하여 부품을 배치한다. 또한 NOT 게이트도 NOT tool(▷)을 이용해 배치한다.



NOT 게이트와 AND 게이트 사이에 공간을 두는 것이 배선을 위해서 유리하다.

이제 입력인 x와 y를 추가해 보자. Input tool (■)을 클릭하여 입력 핀을 배치하고, Output tool (●)을 이용해 OR 게이트의 우측에 출력 핀을 입력한다. OR 게이트와 출력 핀 사이에도 배선을 위한 공간이 필요하다.

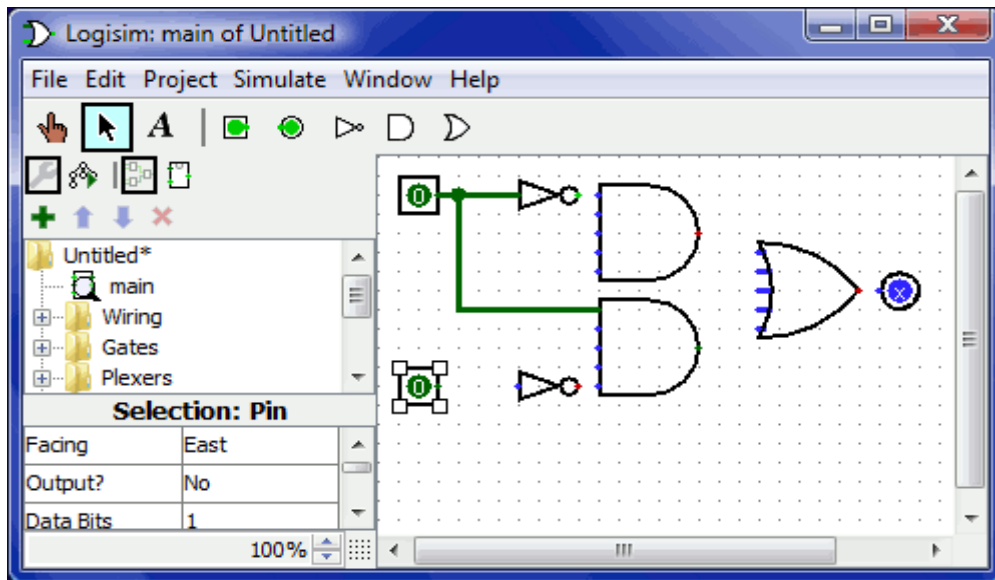


만약 부품 배치가 마음에 들지 않을 경우 Edit tool (☞)을 선택하고 원하는 위치로 부품을 드래그하여 재배치 한다. 삭제를 하려면 Edit menu 에서 Delete 를 선택하거나 그냥 키보드로 Delete 키를 누르면 선택된 부품이 삭제 된다.

배치를 마치게 되면 로지심은 Edit tool 로 되돌아가서 마지막에 선택되었던 부품의 배선작업을 신속하게 할 수 있게 한다. 만약 마지막 선택했던 부품을 추가로 입력하고 싶을 경우 Control-D 를 눌러서 바로 또 다른 부품을 입력할 수 있다.

Step 2: Adding wires (배선 그리기)

부품 배치를 마쳤으면 이제 배선 작업을 하자. Edit Tool(☞)을 선택하고 부품 위에 마우스를 위치하게 되면 작은 녹색 원이 표시가 된다. 이 부분이 배선을 연결하는 부분인데 이 부분에서 마우스를 누른 후 배선을 원하는 부분까지 이동해 마우스를 놓으면 배선이 연결 된다. 로지심은 배선작업에서 상당히 지능적인데 기존 배선 위에다가 추가적으로 현재의 배선의 끝을 위치시키면 자동으로 연결한다. Edit tool 을 이용해 배선을 확장하거나 줄일 수 있다. 로지심에서의 배선은 수평방향과 수직방향 밖에는 지원되지 않는다. NOT 게이트의 왼쪽 입력과 NOT 게이트의 윗쪽 입력을 연결하기 위해 3 개의 wire 를 이용했다 (실제의 배선은 두 개지만 이를 그리기 위해 3 개의 선을 사용)



기존의 배선 위에 T 로 가로점을 연결할 경우 로지심은 자동으로 만나는 점에 점을 표시해주어서 이 부분이 연결된 부분이라는 것을 표시한다.

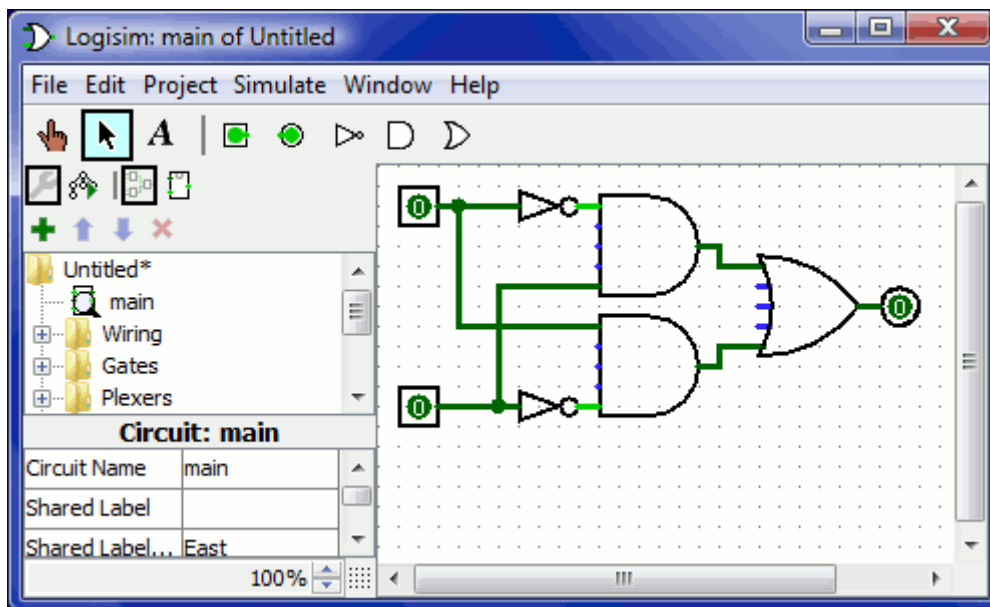
배선을 연결하면 선의 색깔이 파란선 또는 회색으로 표시 될 수 있는데 파란색의 경우 그 포인트의 값이 "unknown" 즉 아직 0, 1 어떤 상태인지를 모른다는 의미이고 회색의 경우에는 다른 부품과 연결이 제대로 되지 않았다는 의미이다. 회로를 다 그린 후에는 이런 파란색이나 회색의 선이 하나라도 보이면 안된다. (OR 게이트의 연결 안된 핀들은 파란색인데 이것은 무방하다)

참고로 각 선의 색깔에 따른 의미는 다음과 같이 정리 된다

- ① 어두운 초록색 : 1 비트 논리 0
- ② 밝은 초록색 : 1 비트 논리 1
- ③ 파란색 : 1 비트의 값을 가지고 있으나 값이 정해져 있지 않음
- ④ 회색 : 부품(컴포넌트)에 연결되지 않아 선의 비트수가 정해지지 않음
- ⑤ 빨강색 : 값을 정할 수 없을 때
- ⑥ 주황색 : 연결된 부품의 비트 폭이 다른 경우

만약 회로를 다 그린 후에도 파란색의 선이나 회색의 선이 남아 있다면 이는 반드시 제대로 된 자리로 치하여야 한다. 부품에서는 연결점을 의미하는 작은 점이 있는데 이 것을 제대로 연결하면 파란색 점에서 녹색으로 바뀌게 된다. 모든 배선이 성공적으로 연결된 경우 배선은 녹색으로 표시된다.

그리고 사용하지 않는 핀의 연결 부위는 파란색으로 남아 있는데 그냥 무시해도 되고 화면 좌측 하단에 있는 Attribute Table 에서 게이트의 입력 수를 필요한 만큼으로 줄이면 이 점들이 없어진다.

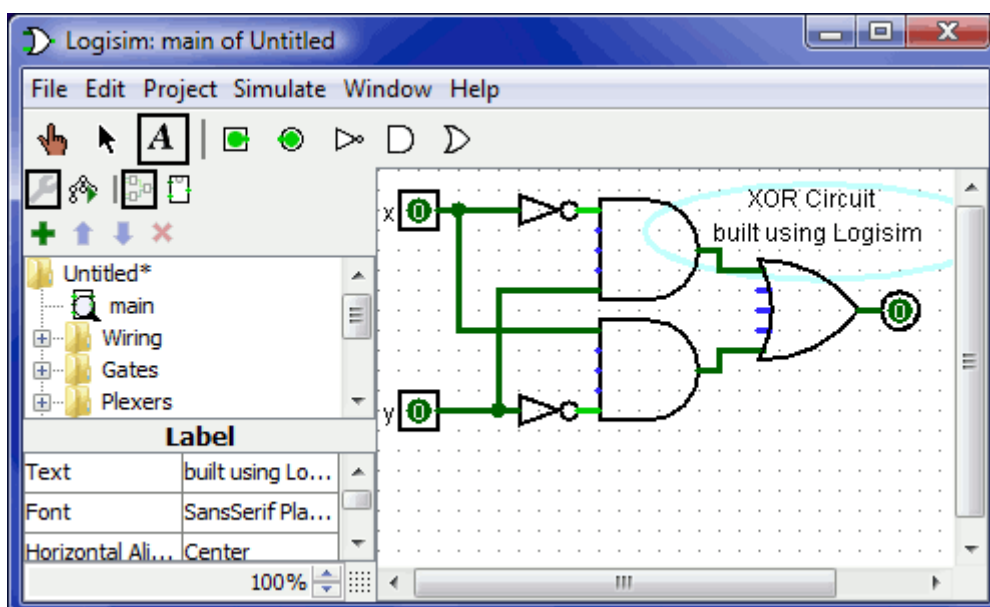


Step 3: Adding text (문자 추가하기)

문자 자체는 회로의 동작과는 큰 관계가 없지만 다른 사람에게 보여 줄 경우 필요하며 또 라벨을 설정하면 시뮬레이션 메뉴에서 로깅(Logging) 기능을 사용할 때 입출력 변수로 사용할 수 있으므로 많은 도움이 된다.

text tool (**A**)을 선택한다. 입력 핀을 클릭하여 라벨을 입력한다. 그런데 글자 삽입을 원하는 장소보다는 입력 핀 자체를 클릭하는 것이 더 좋다.(이렇게 해야 핀이 이동할 때 라벨도 함께 움직이게 된다.) 이런 방식으로 출력 핀도 라벨 입력하면 되고 만약 이미 입력된 라벨을 수정하고 싶으면 그 라벨 위치를 다시 한번 클릭하여 수정을 할 수 있다.

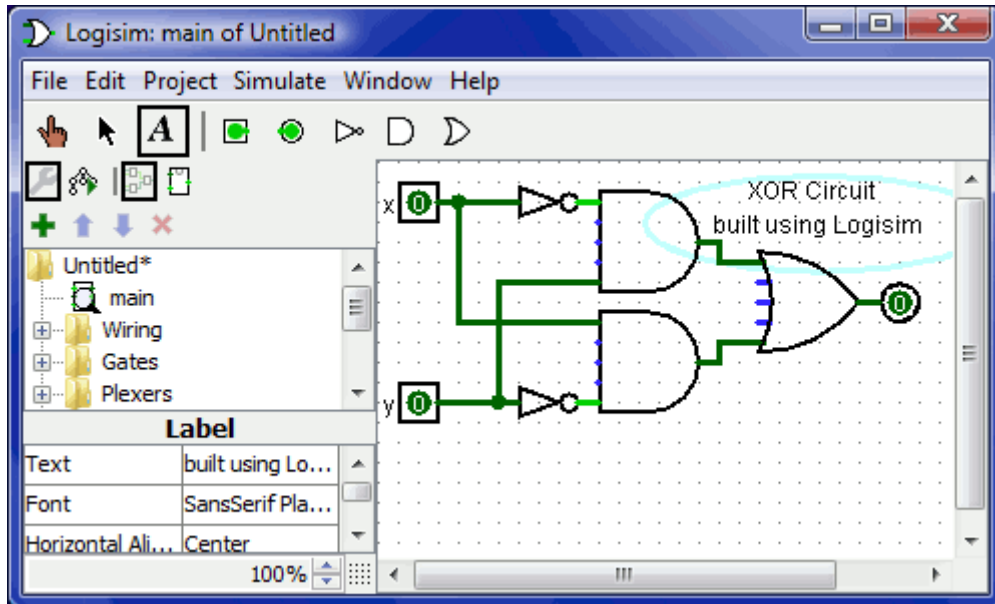
Document 등을 적기 위해서라면 canvas 의 아무 위치나 눌러서 문자열을 입력시킬 수 있다.



Step 4: Testing your circuit (회로 테스트 하기)

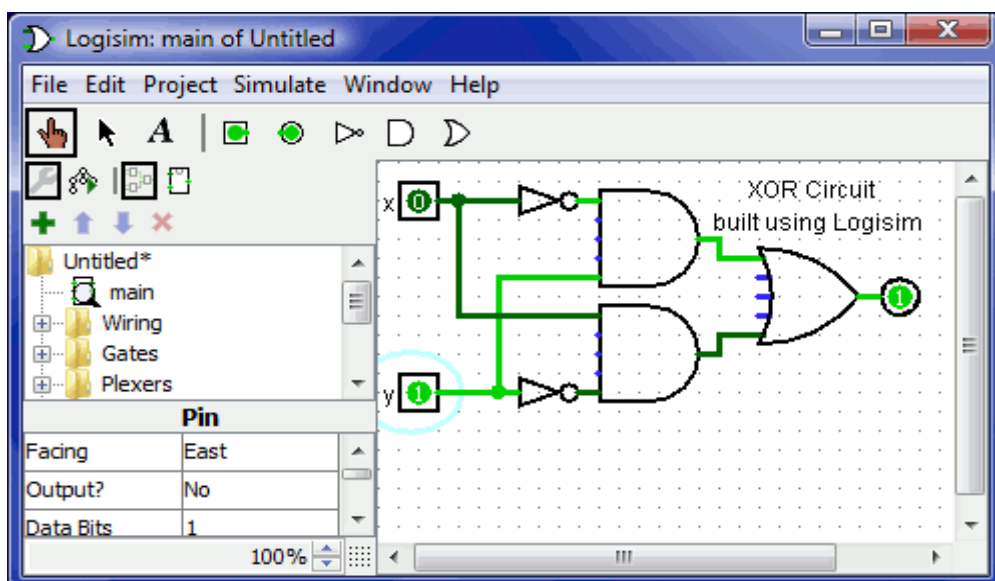
마지막 단계는 우리가 그린 회로가 원하는 것처럼 동작되는지 확인하는 작업이다. 로지심은 이미 회로 시뮬레이션을 진행하고 있는 상태이다.

입력 핀에 설정된 기본값들에 대하여 모든 선들이 어두운 녹색(논리값 1) 또는 밝은 녹색(논리값 0)을 띄게 되며 그 입력 값에 대한 출력도 나타나게 된다.



위 그림에서 두 개의 입력은 모두 0 이고 따라서 출력 결과도 0 이라는 것을 이미 시뮬레이션 해서 보여주고 있다.

이제 입력 조건들을 변경해 보자. poke tool (👉)을 선택해서 변경을 원하는 입력을 클릭한다. 클릭을 할 때 마다 값이 토글 된다는 것을 알 수 있다. 이제 하나의 예로 아래의 입력을 선택해 보자 (poke)



입력 핀의 조건을 바꾸게 되면 로지심은 밝은 녹색선을 통해 1이 입력되어 어떻게 회로 내에서 전달되며 영향을 미치는지를 표시하며, 0 이 입력될 경우 어두운 녹색선으로 경로를 표시해 준다. 출력의 경우도 당연히 자동으로 변경 된다.

이제 진리표의 첫 2 칸의 값($x=y=0$)을 입력하여 출력이 원하는 값(0)으로 나오는지 확인한다. 잘 되었을 경우 나머지 줄도 모두 반복하여 원하는 출력이 모두 나왔을 경우 성공적으로 회로가 구성되었다는 것을 확인할 수 있다.

x	y	$x \text{ XOR } y$
0	0	0
0	1	1
1	0	1
1	1	0

작업을 마치기 전에 지금까지의 내용을 File menu 를 이용해 저장하거나 출력할 수 있다.

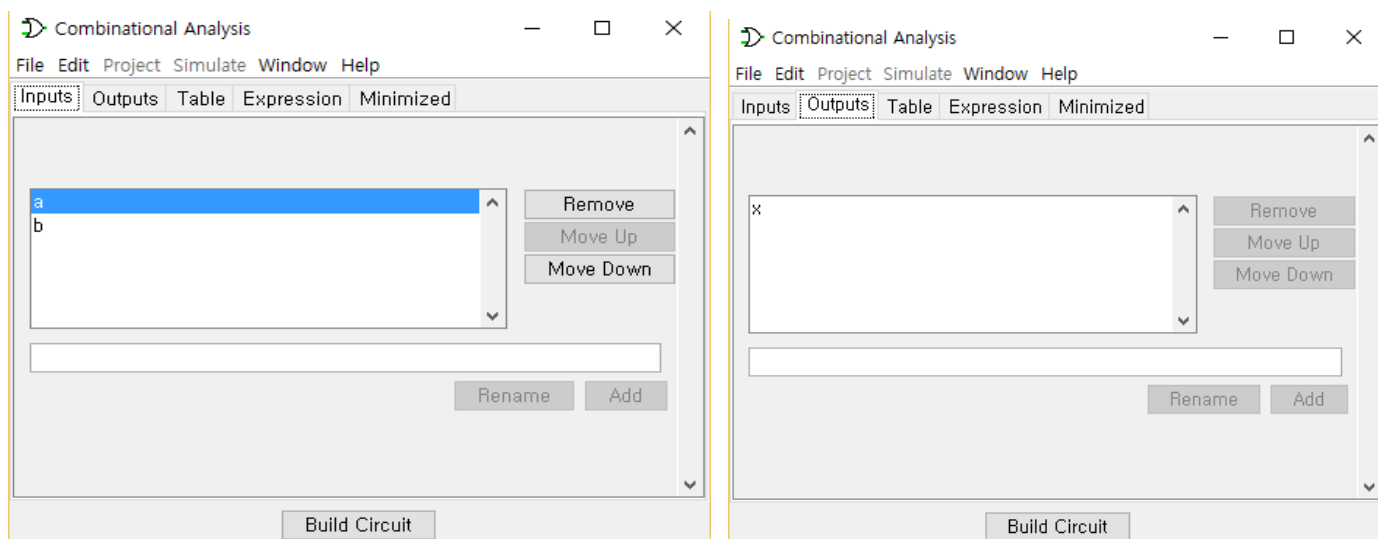
추가 내용 : 분석(Analyze 기능)

이 부분에서 바로 로지심이 매우 유용한 도구임을 알 수 있다

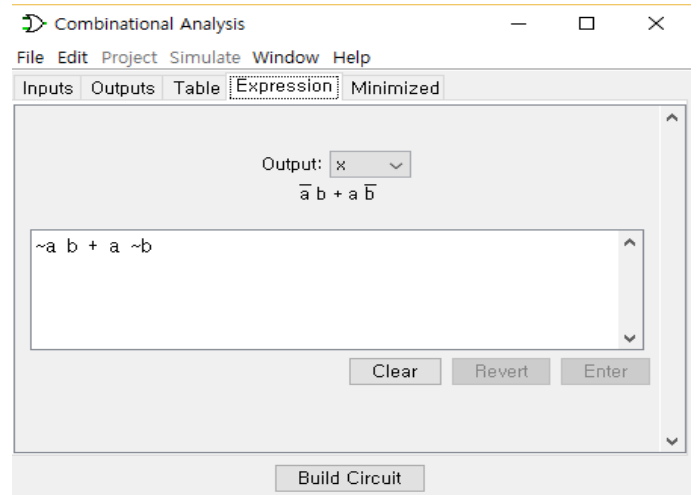
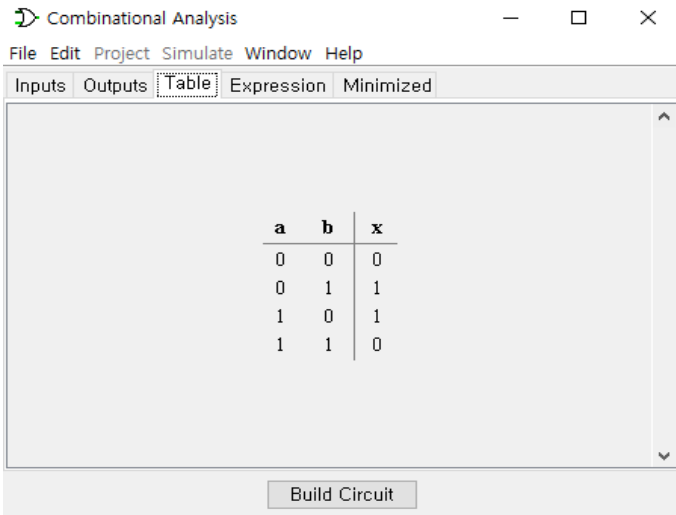
'Project' 메뉴에서 'Analyze Circuit'을 선택해 보자

5 개의 탭이 나오는데 이는 모두 현재 우리가 그린 회로에 대해서 로지심이 분석한 내용이다.

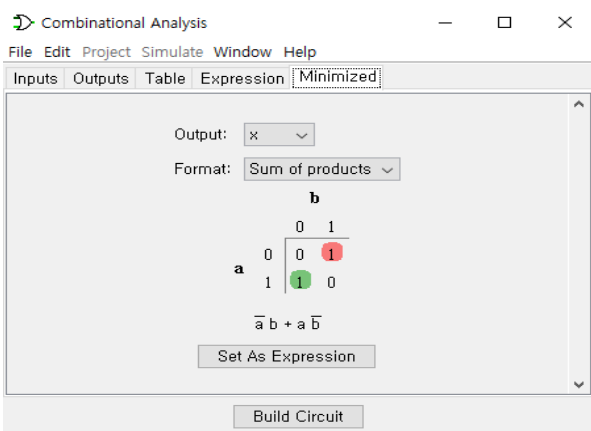
입력 탭, 출력 탭에서는 이름을 바꿀 수 있다



또한 아래 그림에서 보이듯이 우리가 그린 회로에 대한 진리표가 이미 만들어져 있으며 부울 방정식도 이미 만들어진 상태이다.

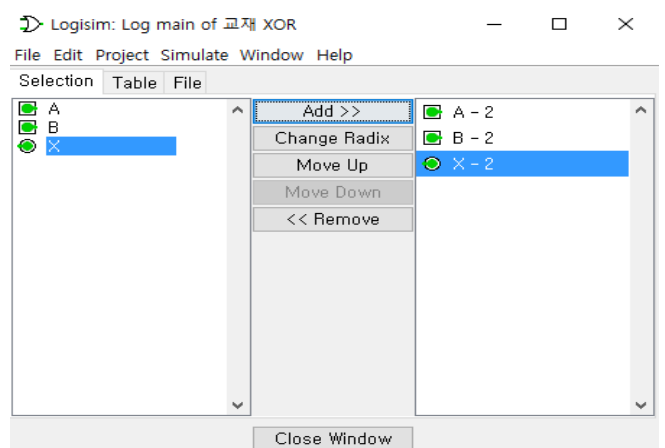
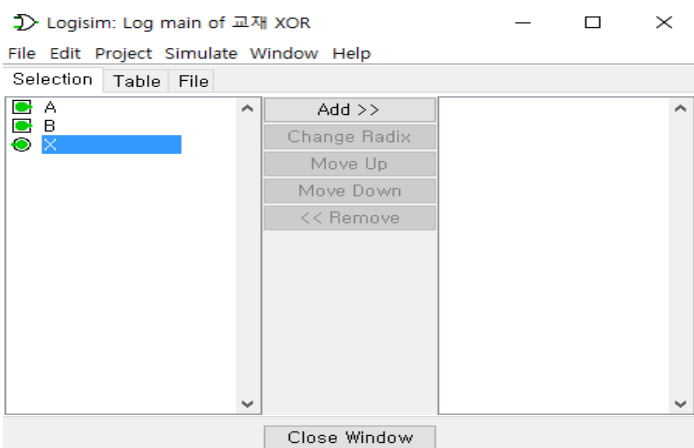


그리고 마지막 5 번째 탭에서는 어떻게 해서 카르노 맵을 간소화 했는지가 표시된다.

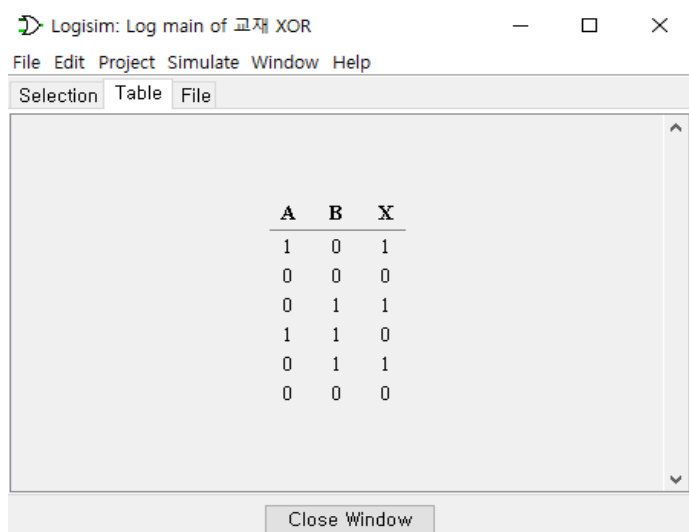


참고로 회로를 먼저 그리고 분석하는 방법도 있지만 거꾸로 'Project' 메뉴에서 'Analyze Circuit'을 선택하고 입력탭, 출력탭 및 table(진료표) 탭을 채우면 나머지 두 탭이 자동으로 산출되며 탭의 밑부분에 있는 'Build Circuit'을 누르면 입력된 값들에 대한 회로를 로지심이 자동으로 그려주는 기능도 있다.

'Simulation' 메뉴에서 'Logging'을 선택하면 다음 그림의 좌측처럼 회로에 대한 입출력 핀들이 나오는데 우리가 시뮬레이션 할 핀들을 우측으로 'Add' 한다.



그리고 메인 화면으로 돌아와 poke tool (👉)을 선택하고 입력 핀의 값들을 바꾸어 주면 이 모든 내용이 기록되며 필요 시에는 저장할 수도 있다.



로지심은 규모가 큰 회로까지 설계, 시뮬레이션이 가능하다. 나머지 부분은 사용자 가이드 또는 Help 메뉴를 이용해 학습하도록 한다.

2. 라이브러리와 속성

로지심에서 생성한 회로들은 라이브러리에 등록하여 더 큰 회로를 구성할 때 하나의 부품으로 불러서 사용할 수 있다. 또한 로지심은 많은 기본 부품들을 라이브러리에 내장해 제공하고 있다.

Explorer pane 에서 기본적으로 제공하는 라이브러리는 다음과 같다

Wiring library

- [Splitter](#)
- [Pin](#)
- [Probe](#)
- [Tunnel](#)
- [Pull Resistor](#)
- [Clock](#)
- [Constant](#)
- [Power/Ground](#)
- [Transistor](#)
- [Transmission Gate](#)
- [Bit Extender](#)

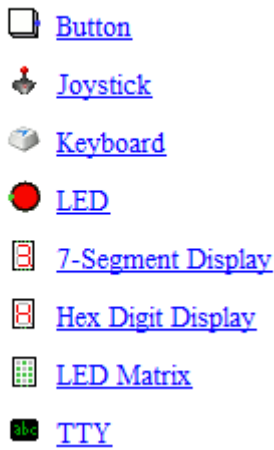
Gates library

- [NOT Gate](#)
- [Buffer](#)
- [AND/OR/NAND/NOR Gate](#)
- [XOR/XNOR/Odd Parity/Even Parity Gate](#)
- [Controlled Buffer/Inverter](#)

Plexers library

- [Multiplexer](#)
- [Demultiplexer](#)
- [Decoder](#)
- [Priority Encoder](#)
- [Bit Selector](#)

Input/Output library



Arithmetic library



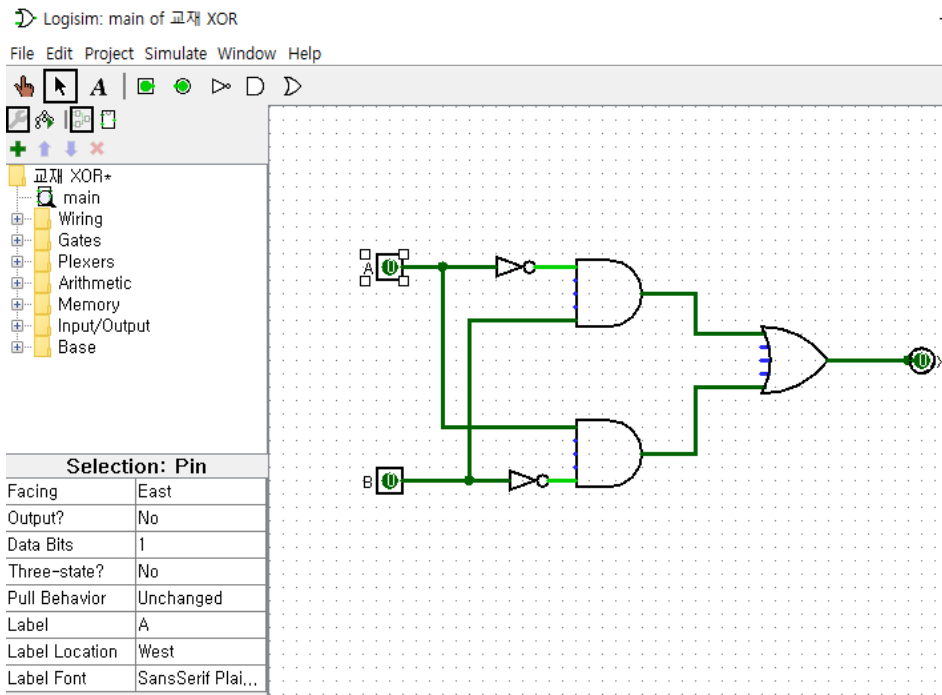
Memory library



Base library



각 부품들은 속성(Attribute)을 가지고 있는데 해당 부품을 선택하면 Attribute Table에 그 부품에 대한 해당 속성이 표시되며 필요 시 이를 수정하여 사용할 수 있다. 핀 배치 방향, 데이터 비트의 폭, 부품의 이름(라벨), 입력 핀 숫자, 폰트 등 다양한 변경이 가능하다.

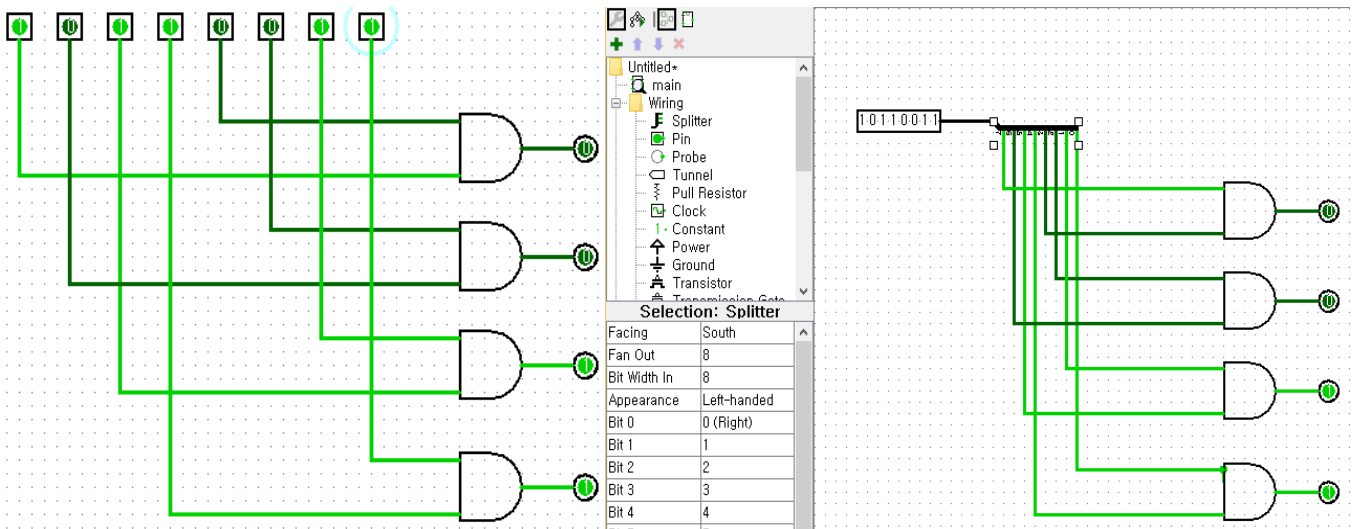


다음은 속성을 이용하여 회로를 쉽게 구현하는 방법 중의 하나의 예이다

8비트 입력이 있으며 이중 상위 4비트와 하위 4비트를 AND 하는 회로를 좌측 그림처럼 그려 보자

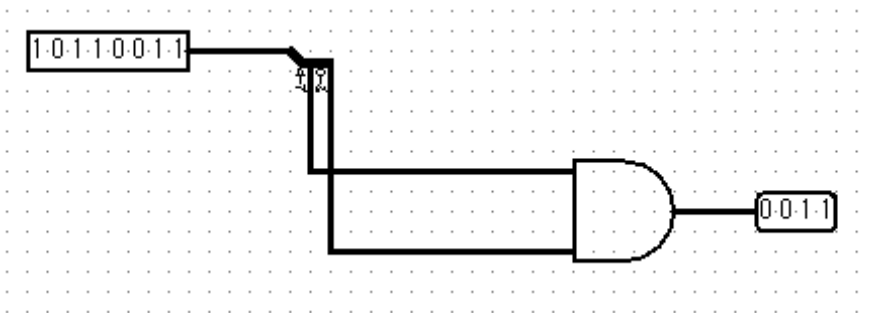
현재 상위 4비트는 1011b, 하위 8비트는 0011b 입력이 들어와 출력이 0011b로 계산된 상태이다.

이를 우측 그림과 같이 입력 핀과 Splitter 핀(팬 아웃 8, South)을 각각 8비트로 변경하여 다음과 같이 간략화가 가능하다. 우측 그림에서 Splitter의 속성을 주의해서 살펴보자.



이를 더욱 간략화 하자면 AND 게이트도 하나만 사용하고 출력도 4비트로 만드는 방법이 있다.

이 방법에서는 Splitter의 팬 아웃을 8개에서 2개로 바꾸어 각각 4비트의 출력이 2개가 나오는 것으로 설정하고, AND 게이트의 속성에서는 입력은 2비트로 하지만 데이터 비트 수를 4로 바꾸고, 오른쪽의 출력 핀도 4비트 입력으로 하면 위의 회로와 동일한 결과를 얻을 수 있다.



Subcircuits의 생성

회로가 복잡해지면 기존에 설계된 모듈들을 불러다가 이를 재활용해 복잡한 회로를 만들 수 있다.

이렇게 로지심에서 제공하지 않지만 사용자가 자주 사용할 기본적인 회로를 만들고 불러 쓸 수 있는데 이러한 사용자 모듈(부품)을 subcircuit 이라 한다.

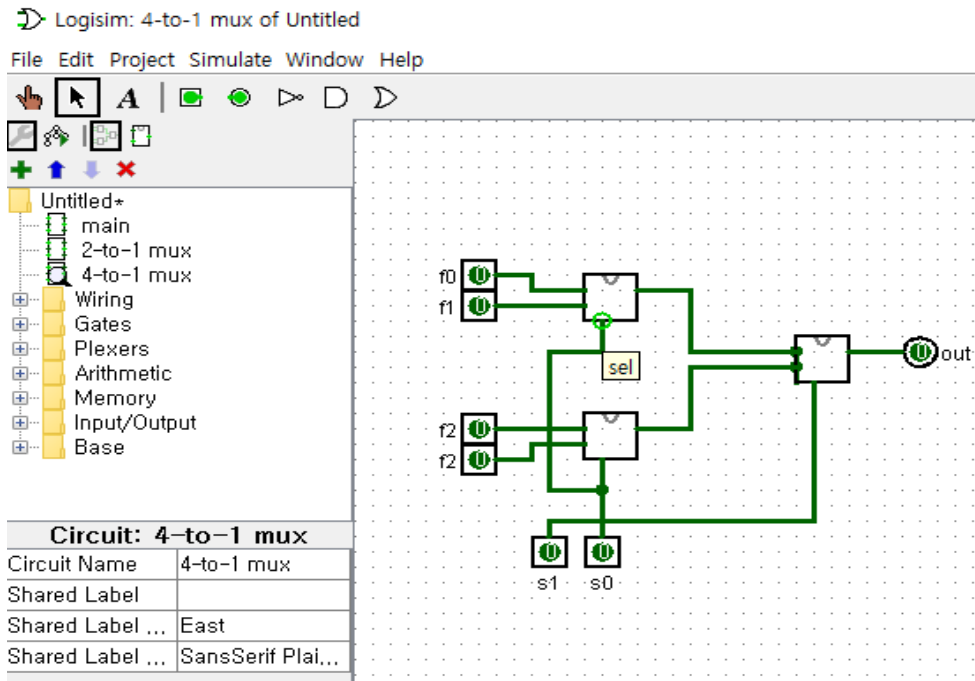
'Project' 메뉴에서 'add circuit'을 선택하고 이름을 지정한다. 예를 들어 '2-to-1 mux'라고 입력하면 Explore pane에 '2-to-1 mux'라는 제목이 나타난다. 그림처럼 원하는 회로를 작성하고 저장(Project asb) 밑의 'save as main circuit'으로 저장)하면 다른 모듈에서 필요 시 이 모듈을 불러서 사용할 수 있다.

Subcircuits의 사용

앞에서 만든 '2-to-1 mux' subcircuit을 이용해 또 다른 '4-to-1 mux' subcircuit을 만드는 방법. 'Project' 메뉴에서 'add circuit'을 선택하고 '4-to-1 mux'로 이름을 지정한다. 이때 Explore pane에 보면 이전에 만든 '2-to-1 mux' subcircuit이 보이는데 선택하고 canvas에서 배치할 곳을 클릭해 배치 한다.

그림처럼 subcircuit의 핀 부분에 입력이 정의되어 있으면 그림처럼 마우스를 가져가면 핀 이름이 표시

된다. (그림에서 sel)



위의 회로에서 2-to-1 mux를 누르면 실제 그려진 회로를 볼 수 있다

Subcircuit 모듈 모양의 편집

'Project' 메뉴에서 'Edit Circuit Apperance'를 선택해 subcircuit의 모양을 변경할 수 있다. 자세한 내용은 생략

- 이상 -