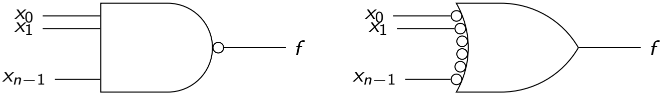
Circuits implemented with only **NAND** and/or **NOR**

- We can implement any circuit with **AND**, **OR**, and **NOT** gates, but we can implement any circuit using *only* **NAND** or **NOR** (or a combination of the two types).
- We might do this for certain considerations — e.g., due to technology considerations where **NAND** and **NOR** gates are cheaper to implement.
- The ability to use only **NAND** or **NOR** follows from the observation that we can make **AND**, **OR**, and **NOT** gates with only **NAND** or **NOR** gates.

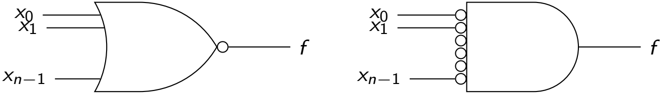
Circuits implemented with only **NAND** and/or **NOR**

- Useful to note that an **NAND** gate performs the same operation as an **OR** gate that has inverted inputs:



$$f = \overline{x_{n-1} \cdots x_1 x_0} = \overline{x_{n-1}} + \cdots + \overline{x_1} + \overline{x_0}$$

- Useful to note that an **NOR** gate performs the same operation as an **AND** gate that has inverted inputs:



$$f = \overline{x_{n-1} + \cdots + x_1 + x_0} = \overline{x_{n-1}} \bullet \cdots \bullet \overline{x_1} \bullet \overline{x_0}$$

Converting 2-level SOP to **NAND** only

- Given a SOP obtaining a NAND only implementation is trivial — apply a double inversion and use the DeMorgan theorem being careful of where we leave the inversions in the final expression.
- Example...

$$
\begin{aligned}
f &= x1!x2+!x1x2+x3 \\
&= !(!f) \\
&= !(!(x1!x2+!x1x2+x3)) \\
&= !(!(x1!x2)!(!x1x2)!(x3)) \\
&= \underbrace{!(\overbrace{!(x1!x2)}^{NAND}\overbrace{!(!x1x2)}^{NAND}\overbrace{!(x3)}^{INV})}_{NAND}
\end{aligned}
$$

Converting 2-level POS to **NOR** only

- Given a POS obtaining a **NOR** only implementation is trivial — apply a double inversion and use the DeMorgan theorem being careful of where we leave the inversions in the final expression.
- Example...

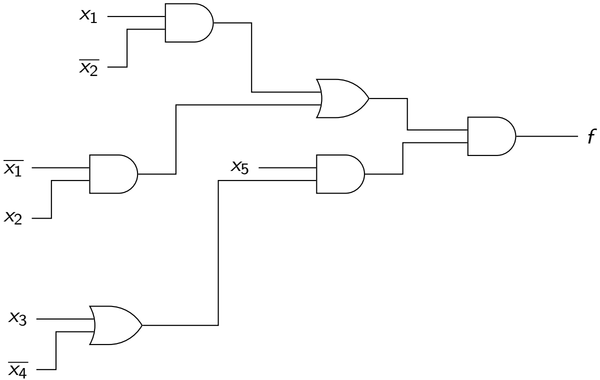
$$
\begin{aligned}
f &= (x1 + x2)(x3 + x4)(x5) \\
&= !(!f) \\
&= !(!((x1 + x2)(x3 + x4)(x5))) \\
&= !(!(x1 + x2)+!(x3 + x4)+!(x5)) \\
&= \underbrace{!(\overbrace{!(x1 + x2)}^{NOR} + \overbrace{!(x3 + x4)}^{NOR} + \overbrace{!(x5)}^{INV})}_{NOR}
\end{aligned}
$$

Multi-level circuit conversions

- For circuits which are not SOP or POS we can still do the conversion (note that this also works for SOP or POS)...
- Insert "double inversions" where necessary to convert gates appropriately.
- We might end up with some left-over inverters, but that's okay.

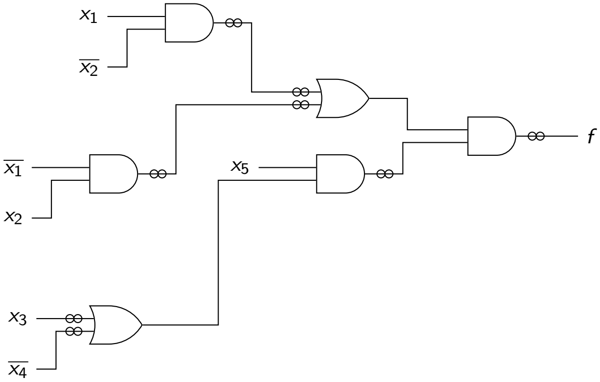
Multi-level circuit conversions — conversion to **NAND**

- To convert to **NAND**: 1. Insert double inversions prior to each **OR**; 2. Insert double inversions after every **AND** gate; 3. Convert gates and cancel out any remaining double inversions.
- Note that there might be some left-over inverters. Note that the use of double inversion ensures that the circuit doesn't change its operation.
- Example: Convert the following to **NAND** only gates.



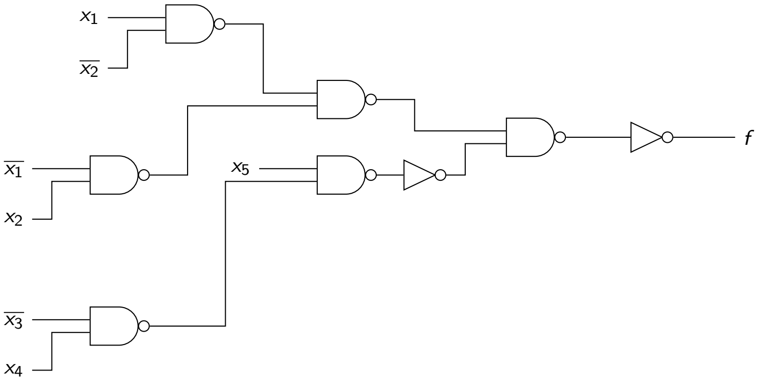
Multi-level circuit conversions — conversion to **NAND**

- To convert to **NAND**: 1. Insert double inversions prior to each **OR**; 2. Insert double inversions after every **AND** gate; 3. Convert gates and cancel out any remaining double inversions.
- Note that there might be some left-over inverters. Note that the use of double inversion ensures that the circuit doesn't change its operation.
- Example: Convert the following to **NAND** only gates.



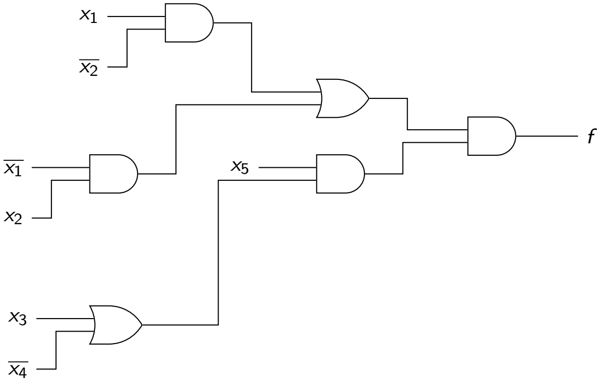
Multi-level circuit conversions — conversion to **NAND**

- To convert to **NAND**: 1. Insert double inversions prior to each **OR**; 2. Insert double inversions after every **AND** gate; 3. Convert gates and cancel out any remaining double inversions.
- Note that there might be some left-over inverters. Note that the use of double inversion ensures that the circuit doesn't change its operation.
- Example: Convert the following to **NAND** only gates.



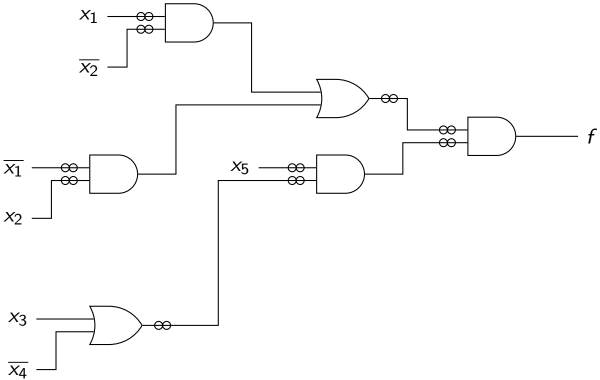
Multi-level circuit conversions — conversion to **NOR**

- To convert to **NOR**: 1. Insert double inversions prior to each **AND**; 2. Insert double inversions after every **OR** gate; 3. Convert gates and cancel out any remaining double inversions.
- Note that there might be some left-over inverters. Note that the use of double inversion ensures that the circuit doesn't change its operation.
- Example: Convert the following to **NOR** only gates.



Multi-level circuit conversions — conversion to **NOR**

- To convert to **NOR**: 1. Insert double inversions prior to each **AND**; 2. Insert double inversions after every **OR** gate; 3. Convert gates and cancel out any remaining double inversions.
- Note that there might be some left-over inverters. Note that the use of double inversion ensures that the circuit doesn't change its operation.
- Example: Convert the following to **NOR** only gates.



Multi-level circuit conversions — conversion to **NOR**

- To convert to **NOR**: 1. Insert double inversions prior to each **AND**; 2. Insert double inversions after every **OR** gate; 3. Convert gates and cancel out any remaining double inversions.
- Note that there might be some left-over inverters. Note that the use of double inversion ensures that the circuit doesn't change its operation.
- Example: Convert the following to **NOR** only gates.