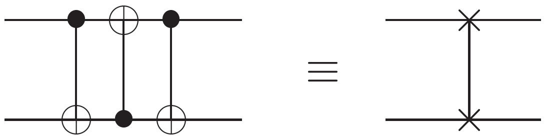

Figure III.11: Diagram for swap (from NC).

C.3 Quantum circuits

¶1. **Quantum circuit:** A *quantum circuit* is a sequential series of quantum transformations on a quantum register.

¶2. The inputs are usually computational basis states (all $|0\rangle$ unless stated otherwise).

¶3. *Quantum circuit diagrams* are drawn with time going from left to right, with the quantum gates crossing one or more "wires" (qubits) as appropriate.
It represents a sequence of unitary operations on a quantum register rather than physical wires.

¶4. **Unique features:** Acyclic: loops (feedback) are not allowed.

¶5. Fan-In (equivalent to OR) is not allowed, since it it not reversible or unitary.

¶6. Fan-Out is not allowed, because it would violate the No-cloning Theorem.
(N.B.: This does not contradict the universality of the Toffoli or Fredkin gates, which are universal only with respect to classical states.)

¶7. **CNOT:** Fig. III.9 (right) shows the symbol for CNOT and its effect.

¶8. **Swap:** The swap operation is defined $|xy\rangle \mapsto |yx\rangle$, or explicitly

$$\sum_{x,y \in \mathbf{2}} |yx\rangle\langle xy|.$$

¶9. We can put three CNOTs in series to swap two qubits (Exer. III.27). It has a special symbol as shown in Fig. III.11.

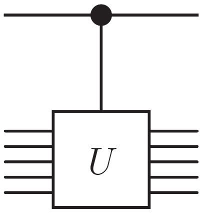

Figure 1.8. Controlled-U gate.

Figure III.12: Diagram for controlled-U (from NC).

¶10. **Controlled-U:** In general, any unitary operator (on any number of qubits) can be controlled (see Fig. III.12). If the control bit is 0, it does nothing, otherwise it does U.

¶11. This is implemented by $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$.
Effectively, the *operators* are entangled.

¶12. **Example:** Suppose the control bit is in superposition, $|\chi\rangle = a|0\rangle + b|1\rangle$.

$$\begin{aligned}
&(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U)|\chi, \psi\rangle \\
&\quad = (|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U)(a|0\rangle + b|1\rangle) \otimes |\psi\rangle \\
&\quad = |0\rangle\langle 0|(a|0\rangle + b|1\rangle) \otimes I|\psi\rangle + |1\rangle\langle 1|(a|0\rangle + b|1\rangle) \otimes U|\psi\rangle \\
&\quad = a|0\rangle \otimes |\psi\rangle + b|1\rangle \otimes U|\psi\rangle \\
&\quad = a|0, \psi\rangle + b|1, U\psi\rangle.
\end{aligned}$$

We have a superposition of entangled outputs.

¶13. Recall that CNOT = controlled X.

¶14. **Conditional or controlled transformation:** If U_0 and U_1 are unitary operators, then we can make the choice between them conditional on a control bit as follows:

$$|0\rangle\langle 0| \otimes U_0 + |1\rangle\langle 1| \otimes U_1.$$

¶15. For example,

$$\text{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X. \tag{III.17}$$

In	Out
$\vert 00\rangle$	$(\vert 00\rangle + \vert 11\rangle)/\sqrt{2} \equiv \vert\beta_{00}\rangle$
$\vert 01\rangle$	$(\vert 01\rangle + \vert 10\rangle)/\sqrt{2} \equiv \vert\beta_{01}\rangle$
$\vert 10\rangle$	$(\vert 00\rangle - \vert 11\rangle)/\sqrt{2} \equiv \vert\beta_{10}\rangle$
$\vert 11\rangle$	$(\vert 01\rangle - \vert 10\rangle)/\sqrt{2} \equiv \vert\beta_{11}\rangle$

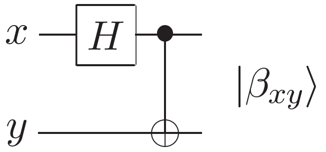

Figure III.13: Quantum circuit for generating Bell states. [from NC fig. 1.12]

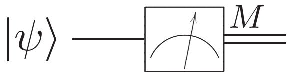

Figure III.14: Symbol for measurement of a quantum state (from NC).

¶16. **Other special gates:** The symbol for the CCNOT gate is show in Fig. III.10,
or with • for top two connections and ⊕ for bottom, representing CCNOT$|x, y, z\rangle = |x, y, xy \oplus z\rangle$,
or put "CCNot" in a box.

¶17. Other operations may be shown by putting a letter or symbol in a box, for example "H" for the Hadamard gate.

¶18. H can be used to generate Bell states (Exer. III.26):

$$\text{CNOT}(H \otimes I)|xy\rangle = |\beta_{xy}\rangle. \tag{III.18}$$

¶19. The circuit for generating Bell states (Eq. III.18) is shown in Fig. III.13.

¶20. **Measurement:** It's also convenient to have a symbol for quantum state measurement, such as Fig. III.14.

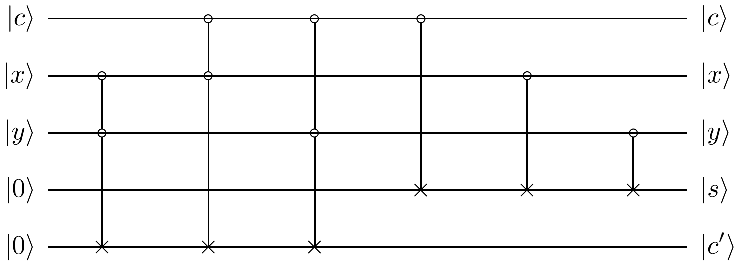

Figure III.15: Quantum circuit for 1-bit full adder [from IQC]. "x and y are the data bits, s is their sum (modulo 2), c is the incoming carry bit, and c' is the new carry bit."

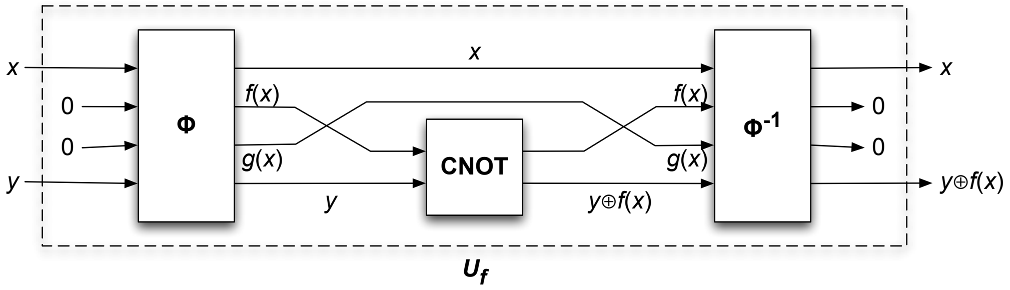

Figure III.16: Quantum gate array for reversible quantum computation.

C.4 Quantum gate arrays

¶1. **Full adder:** Fig. III.15 shows a quantum circuit for a 1-bit full adder.

¶2. As we will see (Sec. C.7), it is possible to construct reversible quantum gates for any classically computable function. In particular the Fredkin and Toffoli gates are universal.

¶3. **Reversibility:** Because quantum computation is a unitary operator, it must be reversible.
You know that an irreversible computation $x \mapsto f(x)$ can be embedded in a reversible computation $(x, c) \mapsto (f(x), g(x))$, where c are suitable constants and $g(x)$ represents the garbage bits.

¶4. Note that throwing away the garbage bits (dumping them in the environment) will collapse the state (equivalent to measurement) by entangling them in the many degrees of freedom of the environment.

¶5. Since NOT is reversible, each 1 bit in c can be replaced by a 0 bit followed by a NOT, so we need only consider $(x, 0) \mapsto (f(x), g(x))$. See Fig. III.16.

¶6. The garbage must be produced in a *standard state* independent of x, "because garbage bits whose value depends upon x will in general destroy the interference properties crucial to quantum computation."

¶7. **Uncomputation:** This is accomplished by uncomputing.
Specifically, perform the computation on four registers (*data, result, workspace, target*):

$$(x, 0, 0, y) \mapsto (x, f(x), g(x), y).$$

Notice that x and y (data and target) are passed through.

¶8. Now use CNOTs to compute $y \oplus f(x)$, where $\oplus$ represents bitwise exclusive-or, in the fourth register:

$$(x, 0, 0, y) \mapsto (x, f(x), g(x), y \oplus f(x)).$$

¶9. Now we uncompute f, but since the data and target registers are passed through, we get $(x, 0, 0, y \oplus f(x))$.
Ignoring the result and workspace registers, we write

$$(x, y) \mapsto (x, y \oplus f(x)).$$

¶10. **Quantum gate array:** Therefore, for any computable $f : \mathbf{2}^m \to \mathbf{2}^n$, there is a reversible *quantum gate array* $U_f : \mathcal{H}^{m+n} \to \mathcal{H}^{m+n}$ such that for $x \in \mathbf{2}^m$ and $y \in \mathbf{2}^n$,

$$U_f |x, y\rangle = |x, y \oplus f(x)\rangle,$$

See Fig. III.17.

¶11. The first m qubits are called the *data register* and the last n are called the *target register.*

¶12. In particular, $U_f |x, \mathbf{0}\rangle = |x, f(x)\rangle$.

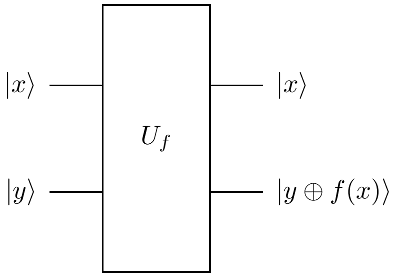

Figure III.17: Computation of function by quantum gate array [from IQC].

C.5 Quantum parallelism

¶1. Since U_f is linear, if it is applied to a superposition of bit strings, it will produce a superposition of the results of applying f to them in parallel (i.e., in the same time it takes to compute it on one vector):

$$U_f(c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \cdots + c_k\mathbf{x}_k) = c_1U_f\mathbf{x}_1 + c_2U_f\mathbf{x}_2 + \cdots + c_kU_f\mathbf{x}_k.$$

¶2. For example,

$$U_f\left(\frac{\sqrt{3}}{2}|\mathbf{x}_1\rangle + \frac{1}{2}|\mathbf{x}_2\rangle\right) \otimes |\mathbf{0}\rangle = \frac{\sqrt{3}}{2}|\mathbf{x}_1, f(\mathbf{x}_1)\rangle + \frac{1}{2}|\mathbf{x}_2, f(\mathbf{x}_2)\rangle.$$

¶3. The amplitude of a result y will be the sum of the amplitudes of all x such that $y = f(x)$.

¶4. **Quantum parallelism:** If we apply U_f to a superposition of all possible 2^m inputs, it will compute a superposition of all the corresponding outputs *in parallel* (i.e., in the same time as required for one function evaluation)!

¶5. The Walsh-Hadamard transformation can be used to produce this superposition of all possible inputs:

$$\begin{aligned} W_m|00\ldots0\rangle &= \frac{1}{\sqrt{2^m}}(|00\ldots0\rangle + |00\ldots1\rangle + \cdots + |11\ldots1\rangle) \\ &= \frac{1}{\sqrt{2^m}}\sum_{\mathbf{x}\in\mathbf{2}^m}|\mathbf{x}\rangle \end{aligned}$$

$$= \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle.$$

In the last line we are obviously interpreting the bit strings as natural numbers.

¶6. Hence,

$$U_f W_m |\mathbf{0}\rangle = U_f \left(\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x, 0\rangle \right) = \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} U_f |x, 0\rangle = \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x, f(x)\rangle.$$

¶7. A single circuit does all 2^m computations simultaneously!

¶8. “Note that since n qubits enable working simultaneously with 2^n states, quantum parallelism circumvents the time/space trade-off of classical parallelism through its ability to provide an exponential amount of computational space in a linear amount of physical space.” [IQC]

¶9. If we measure the input bits, we will get a random value, and the state will be projected into a superposition of the outputs for the inputs we measured.

¶10. If we measure an output bit, we will get a value probabilistically, and a superposition of all the inputs that can produce the measured output.

¶11. Neither of the above is especially useful, so most quantum algorithms transform the state in such a way that the values of interest have a high probability of being measured.

¶12. The other thing we can do is extract common properties of all values of $f(x)$.

¶13. Both of these require different programming techniques than classical computing.