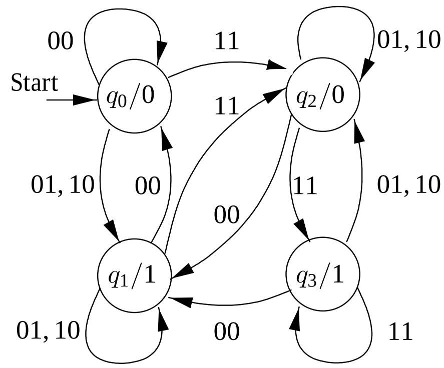

Figure 3.10 A finite-state machine that adds two binary numbers. Their two least significant bits are supplied first followed by those of increasing significance. The output bits represent the sum of the two numbers.

results of this section can be useful. We illustrate this here for binary addition by exhibiting small and shallow circuits for the adder FSM of Fig. 3.10. The circuit simulation for this FSM produces the carry-lookahead adder circuit of Section 2.7. In this section we use matrix multiplication, which is covered in Chapter 6.

The new method is based on the representation of the function $f_M^{(T)} : Q \times \Sigma^T \mapsto Q \times \Psi^T$ computed in T steps by an FSM $M = (\Sigma, \Psi, Q, \delta, \lambda, s, F)$ in terms of the set of **state-to-state mappings** $S = \{h : Q \mapsto Q\}$ where S contains the mappings $\{\Delta_x : Q \mapsto Q \mid x \in \Sigma\}$ and Δ_x is defined below.

$$\Delta_x(q) = \delta(q, x) \tag{3.1}$$

That is, $\Delta_x(q)$ is the state to which state q is carried by the input letter x.

The FSM shown in Fig. 3.10 adds two binary numbers sequentially by simulating a ripple adder. (See Section 2.7.) Its input alphabet is $\mathcal{B}^2$, that is, the set of pairs of 0's and 1's. Its output alphabet is $\mathcal{B}$ and its state set is $Q = \{q_0, q_1, q_2, q_3\}$. (A sequential circuit for this machine is designed in Section 3.3.) It has the state-to-state mappings shown in Fig. 3.11.

Let $\odot : S^2 \mapsto S$ be the operator defined on the set S of state-to-state mappings where for arbitrary $h_1, h_2 \in S$ and state $q \in Q$ the operator $\odot$ is defined as follows:

$$(h_1 \odot h_2)(q) = h_2(h_1(q)) \tag{3.2}$$

q	$\Delta_{0,0}(q)$
q_0	q_0
q_1	q_0
q_2	q_1
q_3	q_1

q	$\Delta_{0,1}(q)$
q_0	q_1
q_1	q_1
q_2	q_2
q_3	q_2

q	$\Delta_{1,0}(q)$
q_0	q_1
q_1	q_1
q_2	q_2
q_3	q_2

q	$\Delta_{1,1}(q)$
q_0	q_2
q_1	q_2
q_2	q_3
q_3	q_3

Figure 3.11 The state-to-state mappings associated with the FSM of Fig. 3.10.

The state-to-state mappings in S will be obtained by composing the mappings $\{\Delta_x : Q \mapsto Q \,|\, x \in \Sigma\}$ using this operator.

Below we show that the operator $\odot$ is **associative**, that is, $\odot$ satisfies the property $(h_1 \odot h_2) \odot h_3 = h_1 \odot (h_2 \odot h_3)$. This means that for each $q \in Q$, $((h_1 \odot h_2) \odot h_3)(q) = (h_1 \odot (h_2 \odot h_3))(q) = h_3(h_2(h_1(q)))$. Applying the definition of $\odot$ in Equation (3.2), we have the following for each $q \in Q$:

$$\begin{aligned}((h_1 \odot h_2) \odot h_3)(q) &= h_3((h_1 \odot h_2)(q)) \\ &= h_3(h_2(h_1(q))) \\ &= (h_2 \odot h_3)(h_1(q)) \\ &= (h_1 \odot (h_2 \odot h_3))(q)\end{aligned} \tag{3.3}$$

Thus, $\odot$ is associative and $(S, \odot)$ is a semigroup. (See Section 2.6.) It follows that a prefix computation can be done on a sequence of state-to-state mappings.

We now use this observation to construct a shallow circuit for the function $f_M^{(T)}$. Let $\boldsymbol{w} = (w_1, w_2, \dots, w_T)$ be a sequence of T inputs to M where w_j is supplied on the jth step. Let $q^{(j)}$ be the state of M after receiving the jth input. From the definition of $\odot$ it follows that $q^{(j)}$ has the following value where s is the initial state of M:

$$q^{(j)} = (\Delta_{w_1} \odot \Delta_{w_2} \odot \cdots \odot \Delta_{w_j})(s)$$

The value of $f_M^{(T)}$ on initial state s and T inputs can be represented in terms of $\boldsymbol{q} = (q^{(1)}, \dots, q^{(T)})$ as follows:

$$f_M^{(T)}(s, \boldsymbol{w}) = \left(q^{(n)}, \lambda(q^{(1)}), \lambda(q^{(2)}), \dots, \lambda(q^{(T)})\right)$$

Let $\boldsymbol{\Lambda^{(T)}}$ be the following sequence of state-to-state mappings:

$$\boldsymbol{\Lambda^{(T)}} = (\Delta_{w_1}, \Delta_{w_2}, \dots, \Delta_{w_T})$$

It follows that $\boldsymbol{q}$ can be obtained by computing the state-to-state mappings $\Delta_{w_1} \odot \Delta_{w_2} \odot \cdots \odot \Delta_{w_j}$, $1 \le j \le T$, and applying them to the initial state s. Because $\odot$ is associative, these T state-to-state mappings are produced by the prefix operator $\mathcal{P}_\odot^{(T)}$ on the sequence $\boldsymbol{\Lambda^{(T)}}$ (see Theorem 2.6.1):

$$\mathcal{P}_\odot^{(T)}(\boldsymbol{\Lambda^{(T)}}) = (\Delta_{w_1}, (\Delta_{w_1} \odot \Delta_{w_2}), \dots, (\Delta_{w_1} \odot \Delta_{w_2} \odot \dots \odot \Delta_{w_T}))$$

Restating Theorem 2.6.1 for this problem, we have the following result.

THEOREM 3.2.1 *For $T = 2^k$, k an integer, the T state-to-state mappings defined by the T inputs to an FSM M can be computed by a circuit over the basis $\Omega = \{\odot\}$ whose size and depth satisfy the following bounds:*

$$\begin{aligned}C_\Omega\left(\mathcal{P}_\odot^{(T)}\right) &\le 2T - \log_2 T - 2 \\ D_\Omega\left(\mathcal{P}_\odot^{(T)}\right) &\le 2 \log_2 T\end{aligned}$$