

Notes & Elaborations on Mermin's (2007) *Quantum Computer Science: An Introduction*

Warren (Hoss) Craft

Summer 2019

EQ 1.35 (SWAP operator in terms of projections and flips)

On pg 12, Mermin [1] notes that the SWAP operator S_{ij} can be written as:

$$S_{ij} = \mathbf{n}_i \mathbf{n}_j + \tilde{\mathbf{n}}_i \tilde{\mathbf{n}}_j + (\mathbf{X}_i \mathbf{X}_j)(\mathbf{n}_i \tilde{\mathbf{n}}_j + \tilde{\mathbf{n}}_i \mathbf{n}_j) \quad (1.35)$$

and he goes on to remark that:

At the risk of belaboring the obvious, I note that (1.35) acts as the swap operator because if both Cbits are in the state $|1\rangle$ (so swapping their states does nothing) then only the first term in the sum acts (*i.e.* each of the other three terms gives 0) and multiplies the state by 1; if both Cbits are in the state $|0\rangle$, only the second term acts and again multiplies the state by 1; if Cbit i is in the state $|1\rangle$ and Cbit j is in the state $|0\rangle$, only the third term acts and the effect of flipping both Cbits is to swap their states; and if Cbit i is in the state $|0\rangle$ and Cbit j is in the state $|1\rangle$, only the fourth term acts and the effect of the two X s is again to swap their states.

Mermin's comment that "each of the other three terms gives 0", for example, can be confusing, and the interpretation of the summation operations could use some clarification as well. To see more clearly what is happening in (1.35), it can help to consider the SWAP operator S_{01} (and its equivalent in terms of the projections and flips as shown in Eq (1.35)) applied in the concrete case of a 2 Cbit system $|xy\rangle$.

Generally, we have $S_{01}|xy\rangle = |yx\rangle$, and more specifically we have:

$$\begin{aligned} S_{01}|00\rangle &= |00\rangle \\ S_{01}|01\rangle &= |10\rangle \\ S_{01}|10\rangle &= |01\rangle \\ S_{01}|11\rangle &= |11\rangle \end{aligned}$$

That is clear conceptually, but it is also useful to remind ourselves of the underlying algebraic processing using the matrix representations and tensor and matrix multiplication. For example,

$$\begin{aligned} S_{01}|10\rangle &= S_{01}(|1\rangle \otimes |0\rangle) = S_{01}\left(\begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right) \\ &= S_{01} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \\ &= |1\rangle_2 = |01\rangle \end{aligned}$$

The vector/matrix representations are useful in eventually interpreting the right-hand side of (1.35) and Mermin's comments about some terms giving a 0, because the zero vector $\bar{0} = \mathbf{0}$ does not admit of a ket-based representation (*i.e.*, the Cbit $|0\rangle = (10 \dots 0)^T$ is not the same thing as the 2^N -dimensional zero vector $(00 \dots 0)^T$).

Toward a better understanding, then, of the right-hand side of (1.35), let us again consider a 2-Cbit system $|xy\rangle$ and recall the matrix representations of the projections (\mathbf{n} , $\tilde{\mathbf{n}}$) and NOT (or “flip”) \mathbf{X} operators:

$$\begin{aligned}\mathbf{n} &= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\ \tilde{\mathbf{n}} &= \mathbf{1} - \mathbf{n} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\ \mathbf{X} &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\end{aligned}$$

Then we would have:

$$\begin{aligned}\mathbf{n}_1 \mathbf{n}_0 |00\rangle &= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ \mathbf{n}_1 \mathbf{n}_0 |01\rangle &= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ \mathbf{n}_1 \mathbf{n}_0 |10\rangle &= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ \mathbf{n}_1 \mathbf{n}_0 |11\rangle &= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |3\rangle_2 = |11\rangle = \mathbf{1}|11\rangle\end{aligned}$$

and we see what Mermin means when he remarks that the first term in the sum (*i.e.*, the $\mathbf{n}_i \mathbf{n}_j$ term) multiplies the state by 1 when the i th and j th CBits are both $|1\rangle$. We can also see that this concrete example generalizes — by considering what happens, for example, when we have the target states within a larger Cbit system. Suppose we take a 5-Cbit system and consider $\mathbf{n}_3 \mathbf{n}_1$:

$$\begin{aligned}\mathbf{n}_3 \mathbf{n}_1 |00000\rangle &= \mathbf{1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \mathbf{n} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \mathbf{1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \mathbf{n} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \mathbf{1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \mathbf{0}_{32}\end{aligned}$$

where $\mathbf{0}_{32}$ indicates a zero vector (consisting of a column vector of 32 zeroes). Whenever a 2×1 zero vector appears as a factor anywhere in the tensor product, we end up with a column vector of all zeros. Thus $\mathbf{n}_i \mathbf{n}_j$ will always produce a zero vector for any input where the i th and j th states are not both $|1\rangle$. For the case where the i th and j th states are both $|1\rangle$, we get some sense of the generality by again considering the action of $\mathbf{n}_3 \mathbf{n}_1$ on a

5-Cbit system:

$$\begin{aligned}
\mathbf{n}_3 \mathbf{n}_1 |11010\rangle &= \mathbf{1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \mathbf{n} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \mathbf{1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \mathbf{n} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \mathbf{1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\
&= (0\ 1\ 0\ 0\ 0\ 0\ 0)^T \\
&= |26\rangle_5 \\
&= |11010\rangle
\end{aligned}$$

References

- [1] N. David Mermin. *Quantum Computer Science: An Introduction*. Cambridge: Cambridge University Press, 2007.