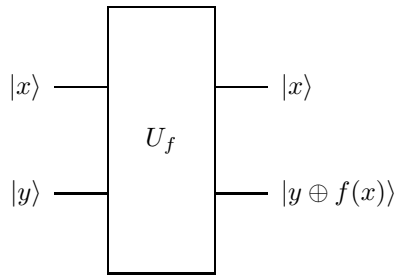

Figure III.17: Computation of function by quantum gate array [from IQC].

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:

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

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

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.