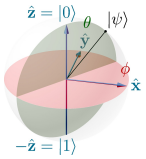
PBP as an Alternative to Quantum

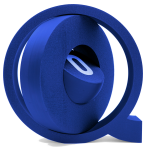
Using ***quantum mechanics phenomena***, quantum computing has the potential to exponentially reduce both the storage space and number of gate actions needed to perform some important computations. Unfortunately, we don't yet know how to build quantum computers that can achieve those goals. **Parallel bit pattern (PBP) computing** is a quantum-inspired computation model that has properties very similar to those of quantum systems, but can be built using conventional logic gates.

An Overview of Quantum Computing

There are several models used for quantum computing, but here we focus on the ***quantum circuits*** model which is used by most quantum computers. Physicists describe the value of a quantum bit, or ***qubit***, as a continuous wave function that defines probability amplitudes for 0 and 1 in terms of coordinates (θ, ϕ) on a ***Bloch sphere***, as diagrammed below and physically simulated by our 3D-printed SPOOKY qubit:



$$|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle$$
$$= \cos(\theta/2)|0\rangle + (\cos\phi + i\sin\phi)\sin(\theta/2)|1\rangle$$
where $0 \le \theta \le \pi$ and $0 \le \phi < 2\pi$



Computationally, that means a qubit can represent the value 0, 1, or a ***superposition*** of both. By having wave functions ***interfere***, multiple qubits can be ***entangled*** so their values are linked. Thus, a quantum computer has the exciting potential to act upon up to 2^n values by performing just a single operation, and a probability density function over 2^n values can be stored in just n qubits.

While quantum systems therefore can perform "parallel computation without parallel hardware," they also have some undesirable attributes:

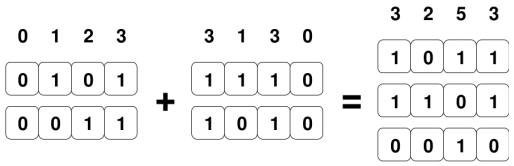
- A superposed value cannot be held indefinitely but suffers ***decoherence***, and measurement immediately collapses the superposition to a randomly-selected single value. There is no way to read the entire probability distribution.
- Only ***reversible logic gates*** can be used and fan-out is not allowed. Thus, additional ***ancilla qubits*** often are needed.
- Computation results are ***nondeterministic*** and subject to errors due to noise.

PBP does not have those problematic attributes...

Introducing Parallel Bit Pattern (PBP) Computing

Where quantum computers encode entangled superpositions using qubits, PBP instead uses ***pbits*** (***pattern bits***). The value of an E-way entangled pbit is an ordered set of 2^E bits; each bit position is an ***entanglement channel***. The bits in corresponding entanglement channels of different pbits behave as entangled values each with a probability of 2^{-E}.

For example, two pbits with the value {0, 1, 2, 3} and two pbits with the value {3, 1, 3, 0} represent the entangled set of value pairs {(0,3), (1,1), (2,3), (3,0)}. Adding these values produces the 2-way entangled 3-pbit value {3, 2, 5, 3}, in which probability of a result of 3 is 50%:



That's exactly how the ***KREQC*** (***Kentucky's Rotationally Emulated Quantum Computer***) demos from SC18 (6-way, 64 bits) and SC19 (16-way, 65,536 bits) work:



It is not how PBP works in the tiny SC22 implementation running on an ESP32 nor on our SC23 ***PBPx4*** and ***KES*** (***Kentucky Entangled Superposition***). Instead of storing and operating on bit vectors, PBP fragments each bit vector into chunks. Only one copy is stored per unique chunk pattern, and many operations can be performed on patterns of chunk descriptors without touching bits. The **PBP as Efficient Bit-Serial SIMD** handout details how this can exponentially reduce the complexity of **`pbit`** operations.

In Our SC26 Exhibit (booth #5637)

The 6 "Q-bit" display of KES not only shows value probabilities using rotation, but also directly displays the bit vector representing the 64 entanglement channels. As the unit executes random quantum gate operations, a 1 bit is represented by a lit LED. However, each 8-bit chunk of each Q-bit's 64-bit vector is colored to show how much work had to be performed to compute it: **white** represents a conventional bit value, **blue** a superposed value not altered by this instruction, **green** a chunk value known symbolically, **yellow** a cached chunk result, and **red** a chunk that actually had to be computed by operating on bits. In other words, despite using no quantum phenomena, *only computation of red chunks does not benefit from an exponential reduction in the number of gate-level operations*.

The not yet operational PBPx4 prototype will directly implement our full 1024-bit chunk PBP model using surplus FPGA hardware, including powerful non-quantum basic operators like entanglement-channel addressing and population count (i.e., read probability).

```
@techreport{sc26pbit,
author={Henry Dietz},
title={{PBP as an Alternative to Quantum}},
institution={{University of Kentucky}},
url={{http://aggregate.org/WHITE/sc26pbit.pdf}},
month={Nov}, year=2026}
```

University of Kentucky

Aggregate.Org UNBRIDLED COMPUTING