Deutsch's Algorithm Dataflow Computing

Abdulwahab Alkharashi
2016 Indiana University, Bloomington

Outline

- Problem Statement
- CPU Code View
- Kennel View
- Discussion

Problem Statement

- We are given a black box in (quantum computer) known as an oracle that implements some function $f\{0,1\}^n \rightarrow f\{0,1\}^n$
- Takes a binary as input and outputs 0 or 1
- Task is to decide whether ***f*** *constant* (0 on all inputs) or *balanced* (1 on all inputs)

CPU Code View

```
void generateInputData(int size, uint32_t *inA, uint32_t *inB,
int8_t *inC){
```

will pass inputs to the core whenever there is data available

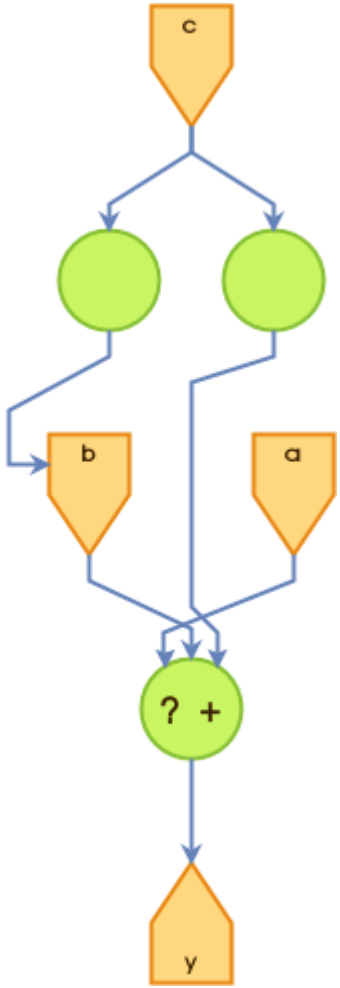
Secant loop

```
int status = 0;
for (int i = 0; i < size; i++){
if (outData[i] != expected[i]) {
fprintf(stderr, "ERROR @ %d, expected %u, got %u\n",i,
expected[i], outData[i]);
status = 1;}}
return status; }
```

Input b will only pass inputs to the core when the current value of input stream c is 1 or a is 0.

Kernel View

```
DFEVar a = io.input("a", dfeUInt(dataWidth));
DFEVar c = io.input("c", dfeBool());
DFEVar b = io.input("b", dfeUInt(dataWidth), c);
DFEVar result = a + (c ? b : 0);
debug.simPrintf("c: %d\n", c);
io.output("y", result, dfeUInt(dataWidth));
```



- Logic: 85.88%
- FFs: 65.00%
- LUTs: 37.85%
- BRAMs: 99.00%
- DSP blocks: 11.10%
- Clock Frequency: 90 MHz

Discussion

Most researchers argue that current quantum programming languages still lack performing a measurement of quantum operations