$\mathcal{NC}$ and $\mathcal{P}$-Completeness

Carry Bit Problem

Given two n-digit binary numerals x and y, when we compute $x + y$, is there be a 'carry' bit in the i^{th} place? (Places are numbered starting with 0 from right to left.)

For example, if we add 1011000101 and 10111011101, there is a carry bit in places 1, 3, 4, 5, 7, 8, 9, 10, and 11.

Problem 1. Prove that the Carry Bit Problem is in $\mathcal{NC}$. Hint: Use the fact that every regular language is in $\mathcal{NC}$. Reduce the carry bit problem to a regular language.

The Circuit Value Problem, or the Boolean Circuit Problem

An instance of the Circuit Value Problem is a sequence of n Boolean assignments.

1. The left side of the i^{th} assignment is the Boolean variable x_i.
2. The right side of the i^{th} assignment is one of the following.
   (a) 0 (false)
   (b) 1 (true)
   (c) x_j for $j < i$
   (d) $!\,x_j$ for $j < i$ (! means 'not')
   (e) $x_j + x_k$ for $j < i$ and $k < i$ (+ means 'or')
   (f) $x_j * x_k$ for $j < i$ and $k < i$ ($*$ means 'and')
3. The answer to an instance of the CVP is the value of x_n.

Trivially, CVP in in $\mathcal{P}$; simply execute the n statements in order.

It is well-known that CVP is $\mathcal{P}$-complete, which implies that if it is in Nick's Class, then $\mathcal{NC} = \mathcal{P}$.

Constant Gap CVP

We define the *constant gap circuit value problem* to be the same as the CVP, except that there is a constant B such that, for each of the conditions listed above, $j > i - B$ and $k > i - B$.

Problem 2. Is the constant gap circuit value problem in $\mathcal{NC}$? Is it $\mathcal{P}$-complete? (Hint: Try small values of B, starting with 2.)