$\mathcal{NC}$ Addition

We assume that we have two length n binary numerals, $\langle x\rangle$ and $\langle y\rangle$. We can compute the binary numeral $\langle x+y\rangle$ in $O(\log n)$ time using $O(n)$ processors; the computation is thus in the class $\mathcal{NC}$.

In our example, $n = 32$. The numerals $\langle x\rangle$ and $\langle y\rangle$ are shown in the first to rows of the matrix below; the leftmost bit is the 31^{st} bit, while the rightmost bit is the 0^{th} bit. The third row shows the **save** bits, the sequence of bits $s_{31},\ldots s_0$ obtained by adding bits of x and y modulo 2.

Our goal is to compute the 33 bit numeral $\langle x+y\rangle$. Let c_i be the carry bit from the $(i-1)^{\text{st}}$ place to the i^{th} place. Since there is no carry bit to the 0^{th} place, we have $c_0 = 0$. The i^{th} bit of $\langle x+y\rangle$ is $s_i + c_i\%2$.

The most difficult part of this problem is computing the carry bits $c_{32}, c_{31},\ldots c_1$ in logarithmic time. You might guess that that is impossible, since a change in the 0^{th} place could effect all carry bits. Our trick is to consider both possibilities at each place simultaneously.

Places and Blocks

We define Place$[i]$ to be consist of the $\imath^{\text{th}}$ bits of x and y. In our example, Place$[0] = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$, Place$[1] = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$, *etc.* For $i \geq j$, let Block$[i,j]$ = Place$[i]$Place$[i-1]\cdots$Place$[j]$.

Each block defines a function $F[i,j]$ from $\{0,1\}$ to $\{0,1\}$ Block$[i,j]$ defines a function from c_j to c_{i+1}. TypeBlock$[i,j] \in \{\text{A},\text{B},\text{C}\}$ as follows

$$\text{Type Block}[i,j] = \begin{cases} \text{A if } c_{i+1} = 0 \\ \text{B if } c_{i+1} = c_j \\ \text{C if } c_{i+1} = 1 \end{cases}$$

We first write the type of each place in the third row of the table using the rules:

$$\begin{aligned} &\text{Type}\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right] = \text{A} \\ &\text{Type}\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right] = \text{Type}\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right] = \text{B} \\ &\text{Type}\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right] = \text{C} \end{aligned}$$

For example, Type Place$[5]$ = A, Type Place$[4]$ = C, and Type Place$[3]$ = Type Place$[2]$ = B, We fill in the rest of the table using *type algebra*, defined by the matrix:

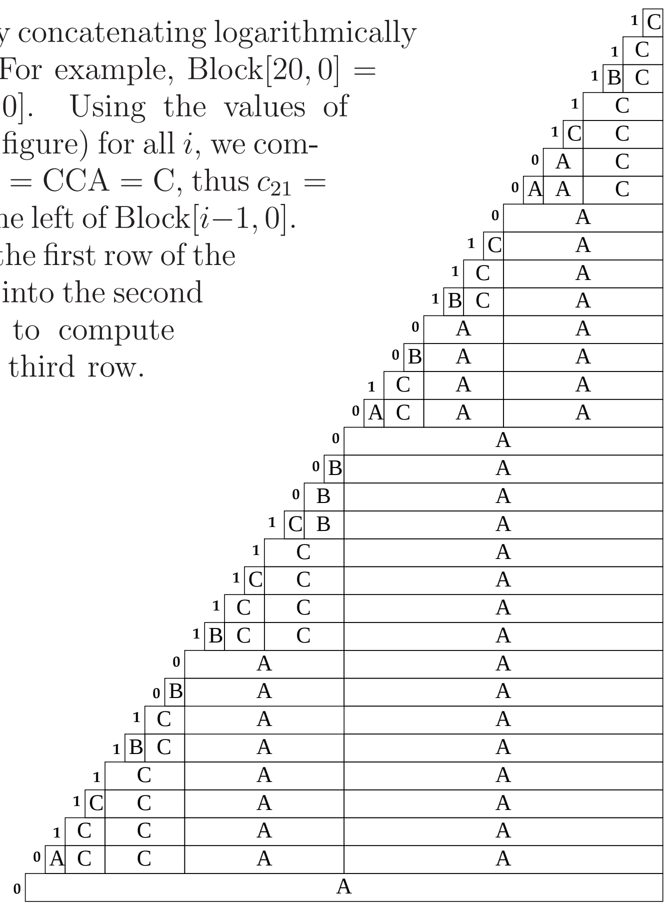
	A	B	C
A	A	A	A
B	A	B	C
C	C	C	C

In five steps, we compute the types of concatenations of blocks of length powers of 2. We compute the types of 16 blocks of size 2, then 8 blocks of size 4, then 4 blocks of size 8, then 2 blocks of size 16, and finally one block of size 32.

x	0	0	1	1	1	0	1	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	1	1	1	0	0	1	1	0	0	1
y	1	0	1	1	0	1	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0	1	0	1	0	0	0	1	0	1	1	1
save	1	0	0	0	1	1	0	1	0	1	1	0	1	0	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	0
	B	A	C	C	B	B	C	B	A	B	B	C	B	C	B	B	B	A	B	B	A	B	B	C	B	A	A	C	B	B	B	C
	A		C		B		C		A		C		C		B		A		B		A		C		A		A		B		C	
	A				C				A				C				A				A				A				C			
	A								A								A								A							
	A																A															
	A																															

Since $c_0 = 0$, we have $c_i = \begin{cases} 0 \text{ if TypeBlock}[i+1,0] \in \{\text{A}, \text{B}\} \\ 1 \text{ if TypeBlock}[i+1,0] = \text{C} \end{cases}$

We compute all Type Block[i-1,0], by concatenating logarithmically many blocks of size a power of 2. For example, $\text{Block}[20,0] = \text{Block}[20,20]\,\text{Block}[19,16]\,\text{Block}[15,0]$. Using the values of Type Block[i-1, 0] (not shown in the figure) for all i, we compute all c_i. For example, $\text{Type}[20,0] = \text{CCA} = \text{C}$, thus $c_{21} = 1$. Each c_i is shown in the figure to the left of $\text{Block}[i-1,0]$. Finally, we write the carry bits into the first row of the table below, then copy the save bits into the second row, and use addition modulo 2 to compute the bits of $x+y$, which are in the third row.



carries	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	1	1	1	1	
save		1	0	0	0	1	1	0	1	0	1	1	0	1	0	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	0
x+y	0	1	1	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	0	0	0	1	0	1	1	0	0	0	0