Hyperthreading and “Almost Amdahl”

Each of the Multiple Cores keeps its own State

1 core, 1 state

2 cores, 2 states

4 cores, 4 states

- Registers
- Program Counter
- Stack Pointer

State

Core

Cache
So, if that’s what Multicore is about, what is **Hyperthreading**?

<table>
<thead>
<tr>
<th>State</th>
<th>Core</th>
<th>Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 core, 1 state</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 core, 2 states, with Hyperthreading</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 cores, 2 states</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 cores, 4 states, with Hyperthreading</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 cores, 4 states</td>
<td></td>
</tr>
</tbody>
</table>

**What is Hyperthreading and what can it Do?**

**Hyperthreading** is when a CPU chip has more states than cores.

In this case, if one thread of execution blocks (waiting for a memory fetch, for instance), then the other thread can resume execution with its state.

If we let $H$ be the fraction of a CPU's capacity that one hyperthread can keep busy, then the remaining unused capacity is $(1-H)$. If another hyperthread can keep $H\%$ of that capacity busy, then that leaves $(1-H)(1-H)$ remaining unused capacity and so on.

If we have $n$ hyperthreads, then the final remaining unused capacity is $(1-H)^n$. The capacity actually in use would then be $1-(1-H)^n$. If one thread can only keep the CPU $H\%$ busy, then the speed-up is potentially:

$$SU = \frac{1-(1-H)^n}{H}$$
If we have \( n \) hyperthreads, then the final remaining unused capacity is \( (1-H)^n \). The capacity actually in use would then be \( 1-(1-H)^n \). If one thread can only keep the CPU \( H \% \) busy, then the speed-up is potentially:

\[
SU = \frac{1-(1-H)^n}{H}
\]

**What is Hyperthreading and what can it Do?**

**A Lidar Application:**
**Four Cores with Two Hyperthreads per Core**

Note that this is upside-down from our usual convention. Sorry. I got this from someone else.