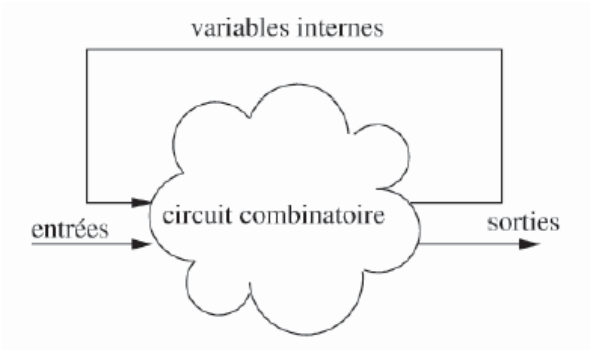
Dedicated Architectures for Signal Processing

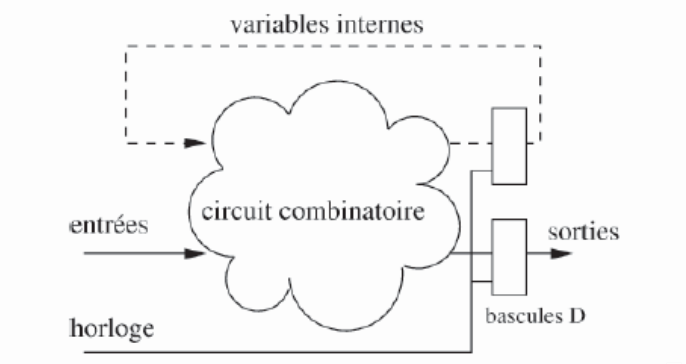
Lirida A. B. Naviner

Institut TELECOM – TELECOM ParisTech
Ecole Nationale Supérieure des Télécommunications
Département Communications et Electronique
46, rue Barrault – 75634 Paris cedex 13
33 (0)1 45 81 76 61 33 (0)1 45 80 40 36 lirida.naviner@telecom-paristech.fr

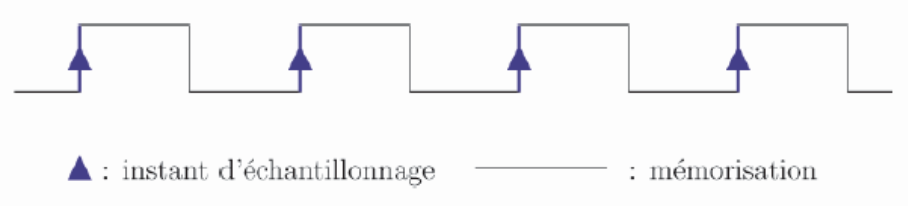
2008–2009

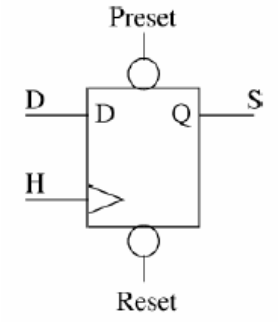
Sequential Synchronous Logic





Sequential Synchronous Logic (cont.)



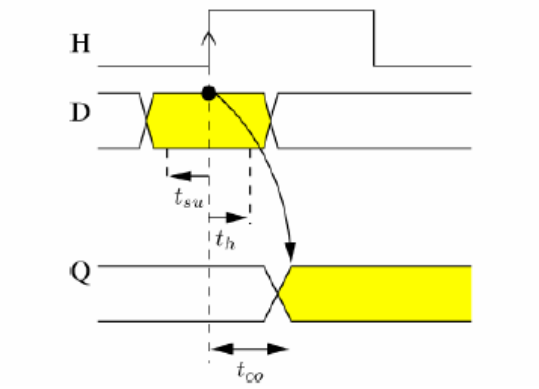


Sequential Synchronous Logic (cont.)

D	H	Preset	Reset	Q	Etat
0	↑	1	1	0	2sampling
1	↑	1	1	1	
X	0	1	1	Q	2storage
X	1	1	1	Q	
X	X	0	1	1	1
X	X	1	0	0	0

Critical Delay

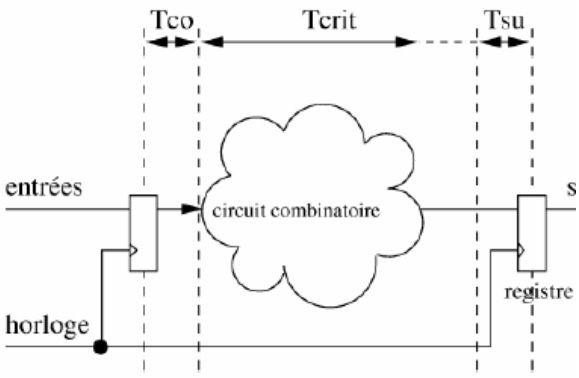
$$t_{ck} > t_{crit} \quad (1)$$



Critical Delay (cont.)

$$t_{ck} > t_{co} + t_{crit} + t_{su} \quad (2)$$

$$t_{ck} > t_{crit} + t_R \quad (3)$$



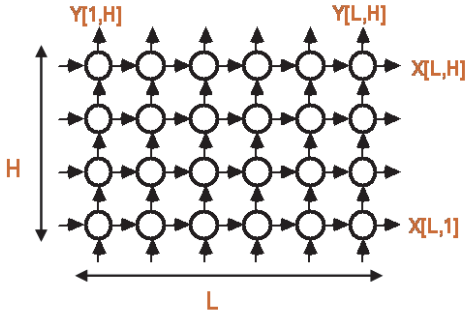
Metrics

- The **area** S, generally given with respect to a reference value (for example, the full adder area S_{FA})
- The **speed**, generally given by maximal frequency of functionning f_{max} (or, equivalently by the minimal clock period t_{ck})
- The **power consumption** C, that is a function of the activity rate, the operation frequency and the area of the operators.
- The **computation power** P_c, given by the ratio between the amount of operations and the time period required to implement these operations.
- The **reliability** R, represented by the probability of a correct output given probability or errors inside the operator.

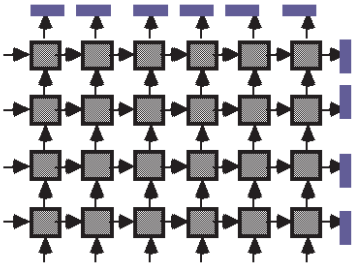
Temporal Resources Distribution

Algorithm

- $L \times H$ basic operations Φ, each Φ is performed by a combinational operator ϕ, with area cost S_ϕ and delay t_ϕ.



Parallel Implementation

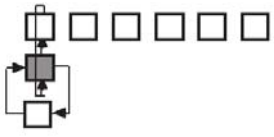


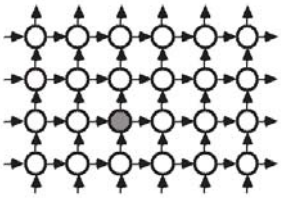
$$t_p = (L + H - 1) \times t_\phi + t_R \quad (4)$$

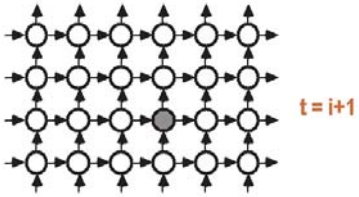
$$S = (L \times H) \times S_\phi + (L + H) \times S_R \quad (5)$$

$$P_c = \frac{(L \times H)}{(L + H - 1) \times t_\phi + t_R} \phi/s \quad (6)$$

Sequential Implementation







$$t_p = t_\phi + t_R \quad (7)$$

$$S = S_\phi + (L + 1) \times S_R \quad (8)$$

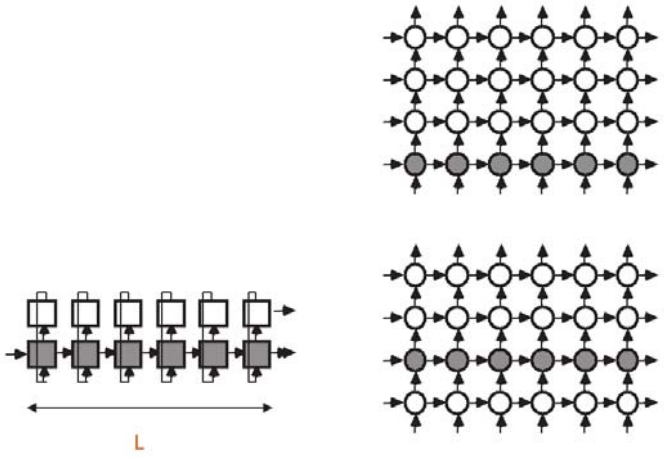
$$P_c = \frac{1}{t_\phi + t_R} \phi/s$$

Mixed Implementation

$$t_p = L \times t_\phi + t_R \quad (10)$$

$$S = L \times (S_\phi + S_R) \quad (11)$$

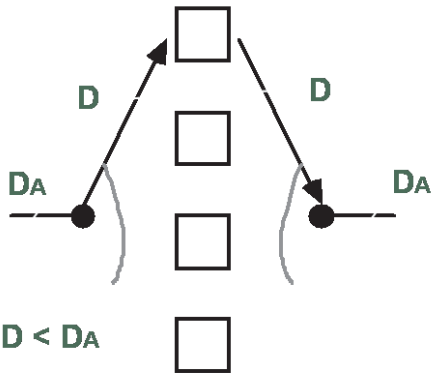
$$P_c = \frac{L}{L \times t_\phi + t_R}\ \phi/s \quad (12)$$

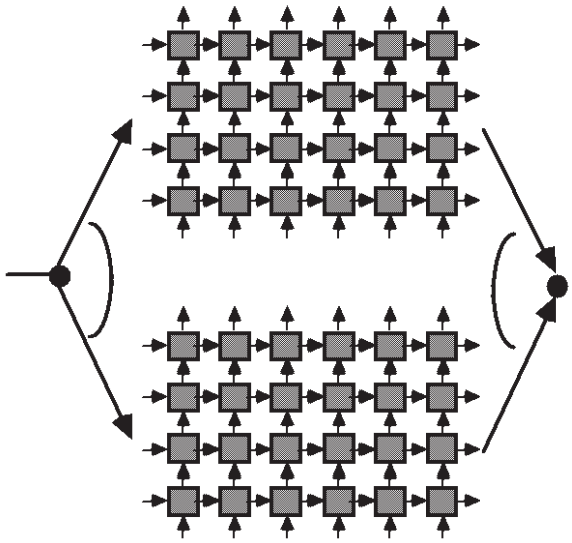


Multiplexing

Application : D_A

Ressources : D

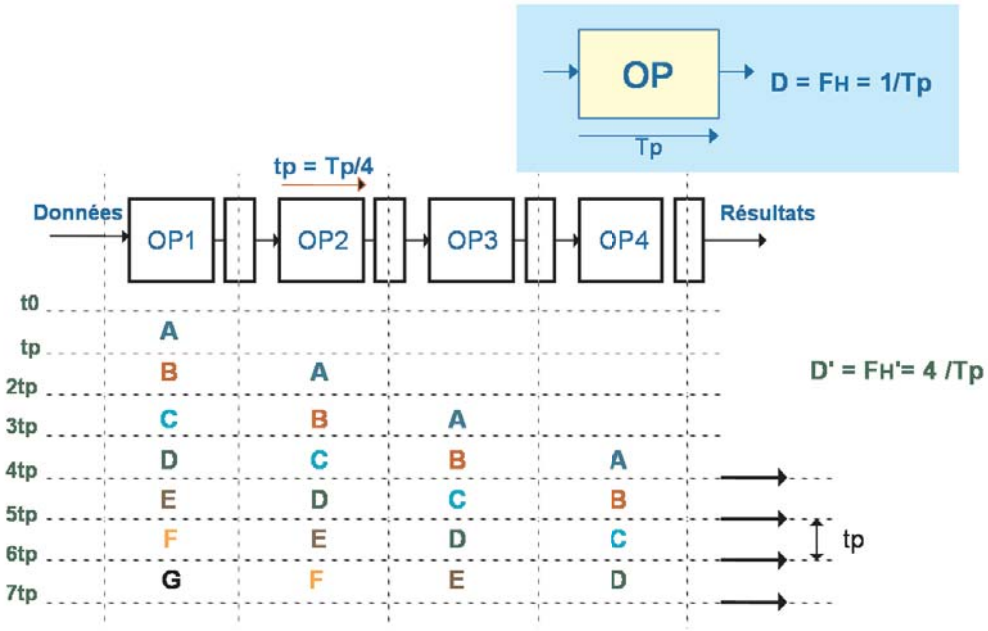




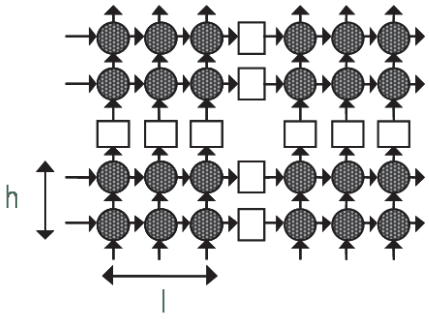
Exemple : n=2

Pipeline

$$D' = f'_{ck} = N/t_p \tag{13}$$



Horizontal/vertical Pipeline

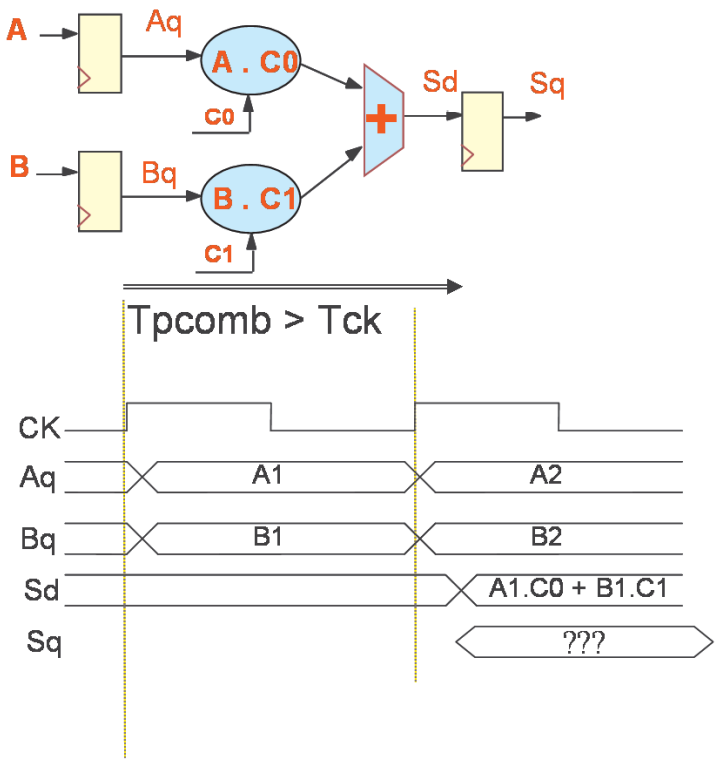


$$t_{ck} = (l + h - 1) \times t_\phi + t_R \tag{14}$$

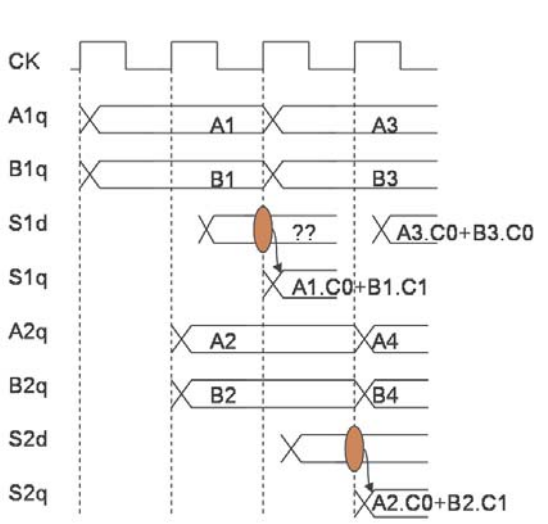
$$S = LH \times S_\phi + \left(\left\lceil \frac{L}{l} \right\rceil H + \left\lceil \frac{H}{h} \right\rceil L \right) \times S_R \tag{15}$$

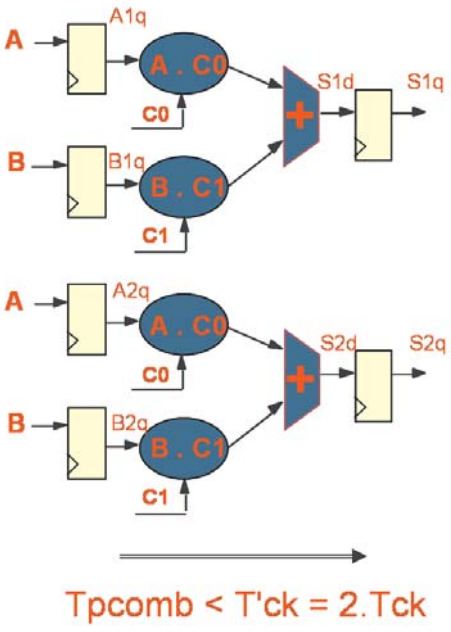
$$P_c = \frac{LH}{(l + h - 1) \times t_\phi + t_R} \phi/s \tag{16}$$

Temporal/spatial Resources Distribution

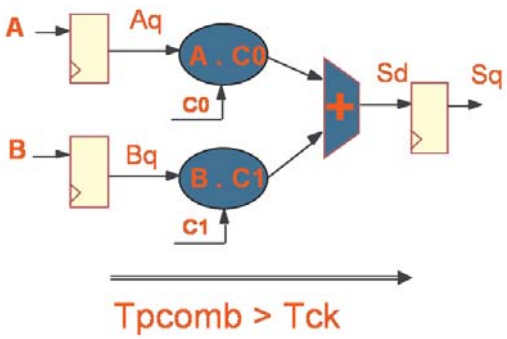


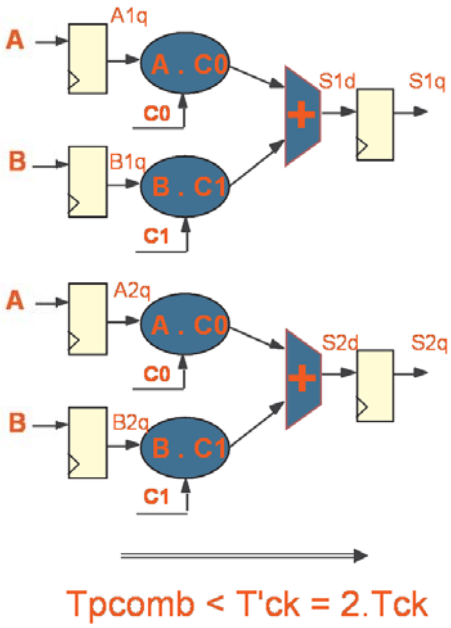
Acceleration with Multiplexing





Acceleration with Multiplexing

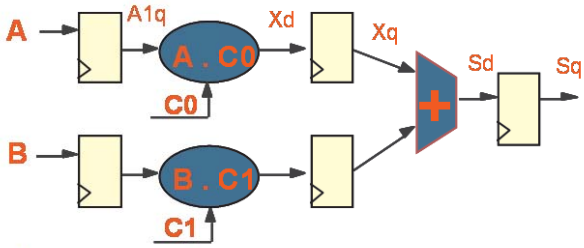


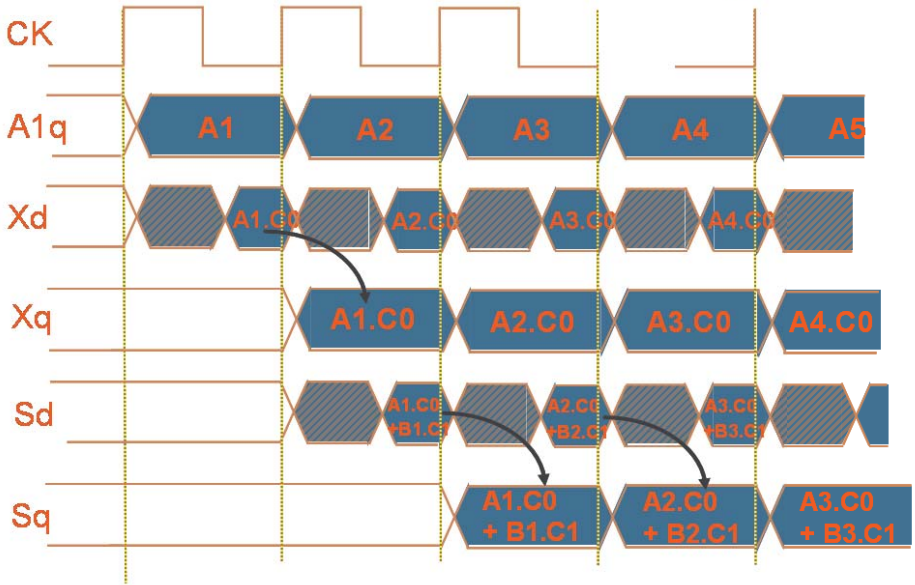


Solution 1:

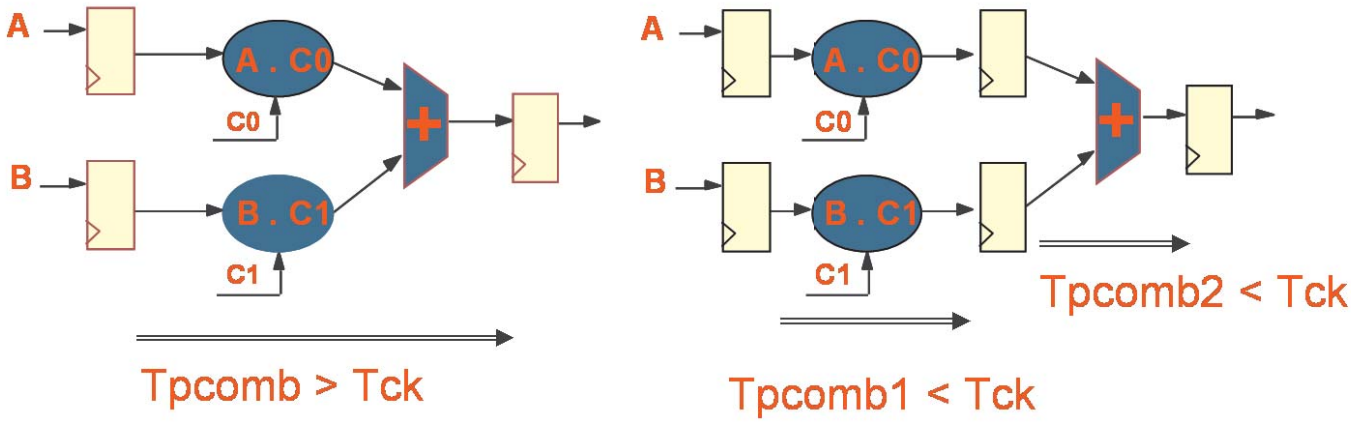
- Hardware x 2
- Same data rate

Acceleration with Pipeline





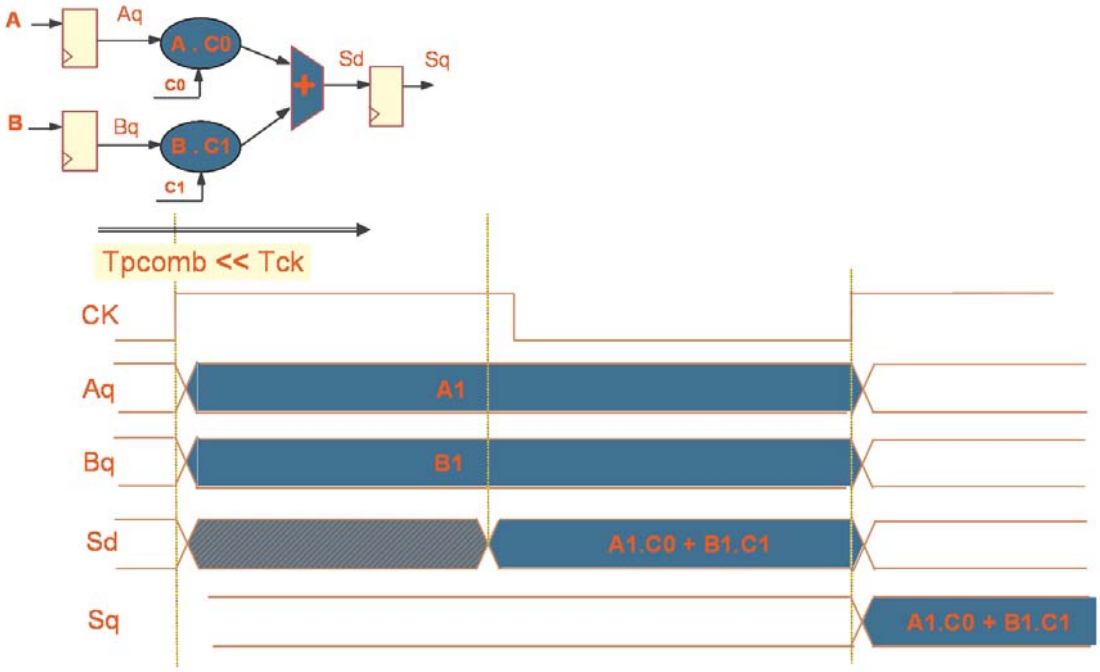
Acceleration with Pipeline



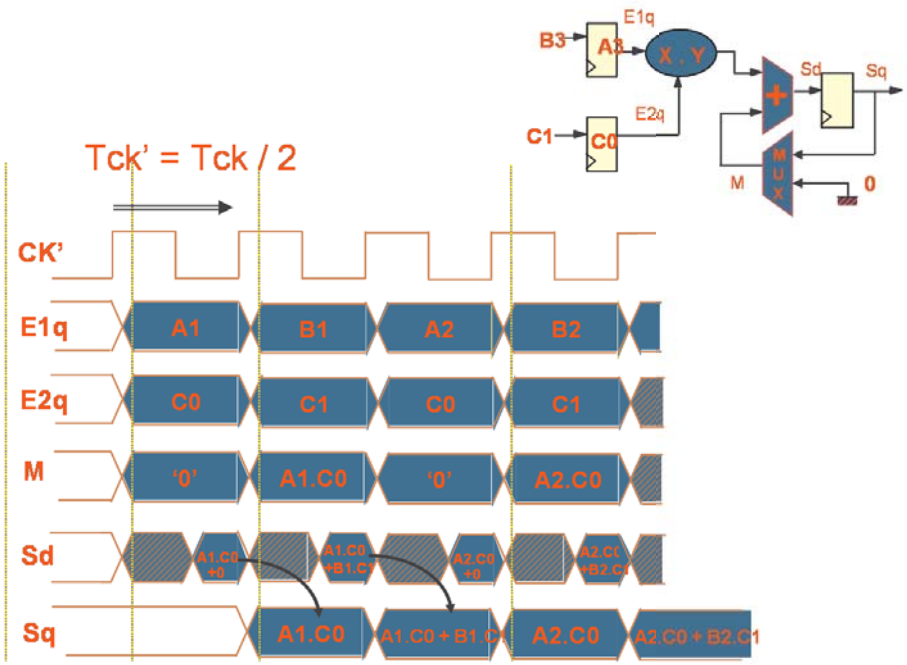
Solution 2:

- To split the combinatorial logic in many equi-temporal stages separated by registers
- Same data rate
- Increased Latency (in CLK periods) but not necessarily the delay (in seconds)

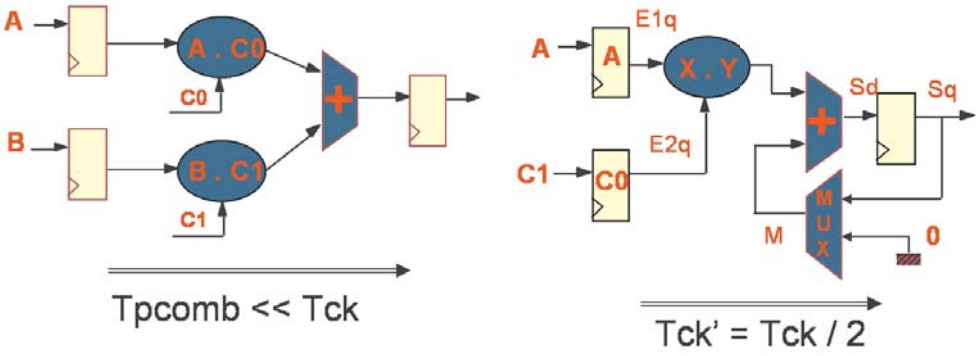
Temporal/spatial Resources Distribution



Saving area with Sequential Calculations



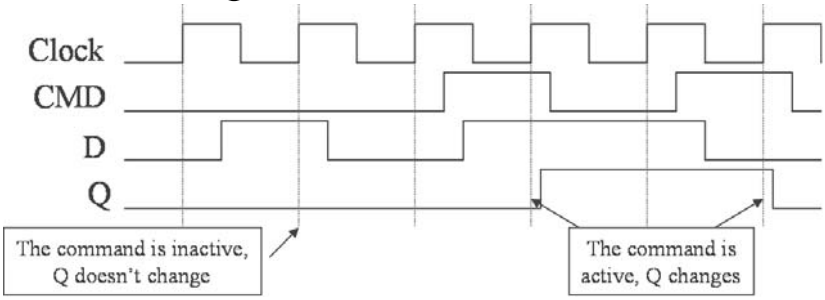
Saving area with Sequential Calculations



- Resource sharing = Hardware saving
- Same data rate
- Same latency

Exercise ✎

A flip-flop D samples its input at each rising edge of the clock signal. Derivate a structure avoiding sampling at some instants specified by a command signal CMD.



Exercise ✎

Consider the output signals given by the equations below. Suppose that input signal x is sampled at frequency F_s (x_n denotes the n-th sample of x).

$$y_n = OP_2(x_n, w_{n-2})$$
$$w_n = OP_2[OP_1(w_{n-2}), OP_1(z_{n-2})]$$
$$z_n = OP_2[(OP_1(y_n), OP_1(w_n)]$$

Consider two combinational operators **OP1** and **OP2** (that perform operations OP_1 and $OP2$, respectively). The operator **OP2** is commutative.

1. Derivate a parallel architecture to give y_n, w_n and z_n (cf. section 1).
2. Suppose propagation delays $t_{OP1} = t_{OP2} = 25$ns. Suppose ideal register delay ($t_R = 0$). Derivate the maximal frequency for an implementation based on the proposed architecture.
3. If the maximal frequency is inferior to 20 MHz, make changes in your architecture in order to accept input signals x at 20MHz.
4. Is it possible to process signal x sampled at 40Hz?

Exercise ✎

Let be the iterative algorithm presented in the graph of the figure 4. An iteration consists on one step of calculation followed by wire multiplexing. The calculation step needs 8 operations OP on a set of 16 operands (D_0 to D_{15}). Consider an application requiring 256 iterations of a such algorithm at each $29\mu s$. Also consider you dispose of registers and fully combinational operators **OP** that perform the operation OP shown in the nodes of the given graph. The operators are caracterized by propagation delay $t_{OP} = 90ns$. The registers are caracterized by propagation delay $t_R = 2.5ns$.

1. Give the maximal operation frequency f_{max} accepted by operator **OP**.
2. Give the computation power required by the application (in OP/s).
3. Let be the implementation of the algorithm based on only one pipelined operator **OP**. How many pipeline barriers are necessaries?
4. Consider the pipelined operator **OP** of previous item. Give the sequencying of operations for two iterations. What are your conclusions?

Exercise ✎