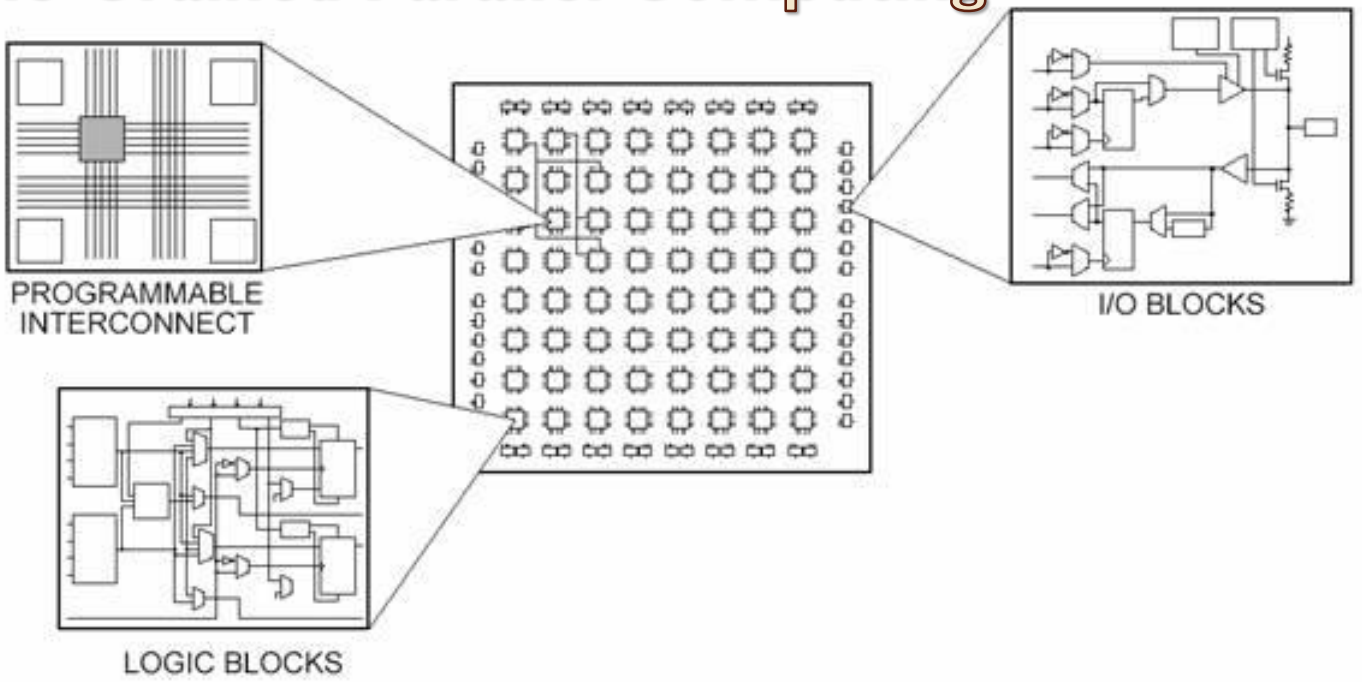
Auto-Configurable Arrays for GCD Computation

Fine-Grained Parallel Computing



Plamen Alexandrov, ISI Master Student '08-09

JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

ISI International School for Informatics HAGENBERG

(Parallel) GCD Algorithms

- Right to left processing
  - Carry propagation
  - Positive/negative sign
  - Parallel/Sequential input (all digits are required)
- Left to right
  - Comparison required to keep operands non-negative
  - Positive/negative sign detection
  - Termination detection
- Dynamic length of operands
  - Vary between different application
  - May vary during the same computation
  - "packing" cells into one lead to other problems

JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

ISI International School for Informatics HAGENBERG

Improved Plus-Minus Algorithm

1. **Preprocessing**:
   - Least-significant common **null bits** are discarded from both operands A and B;
   - If LSB a_0 of A is null, then the operands are interchanged (a_0 always becomes 1);
2. **Reduction** (iterate until B becomes null):
   - If (LSB of B) b_0=0 then B is shifted $(B \leftarrow B/2)$
   - If (LSB of B) b_0=1 then $(A, B) \leftarrow (B, (A \pm B)/4)$ where we have *plus* if $a_1 \neq b_1$ and *minus* otherw.

JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

ISI International School for Informatics HAGENBERG

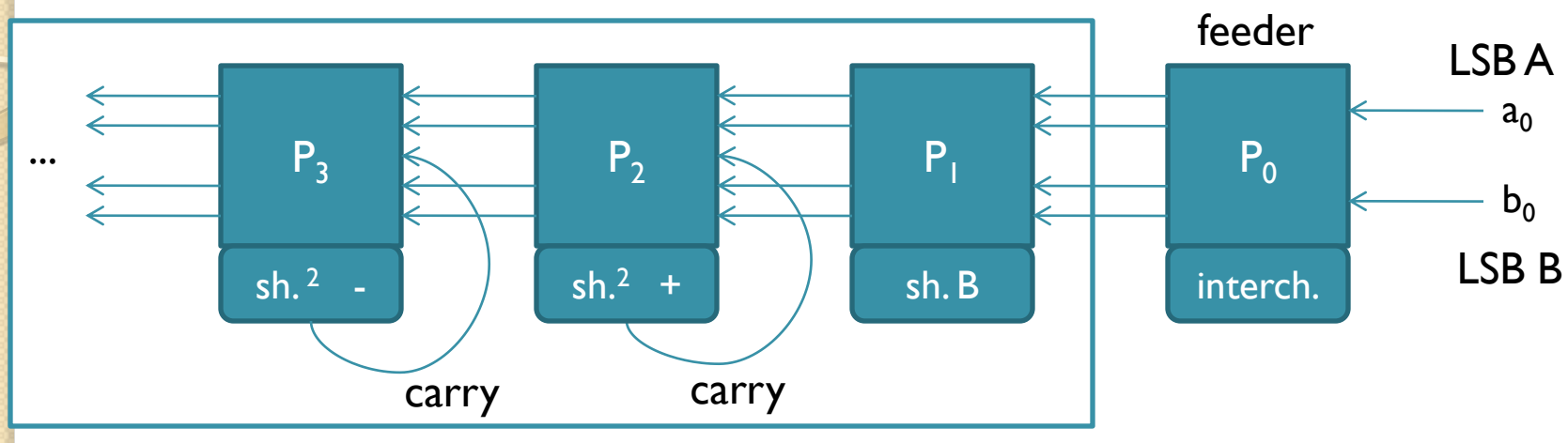
I-PlusMinus Algorithm Benefits

- Based on:
  - GCD(A, B) = GCD(A + B, B);
  - If A and B are *odd* then 4/(A + B) or 4/(A - B);
- Each step consists of:
  - Setting $A'_{k+2}=A_k+A_{k+1}$ or $A'_{k+2}= A_k-A_{k+1}$ $(4/A'_{k+2})$;
  - And double shift of $A_{k+2} = A'_{k+2}/4$;
- It is not important to know which is the largest: A_k or A_{k+1} at each step:
  - No comparison required at each step;
  - Works even for negative operands;
  - No need to keep track of the most significant digits;

JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

ISI International School for Informatics HAGENBERG

Single Pass-Through FPGA Array

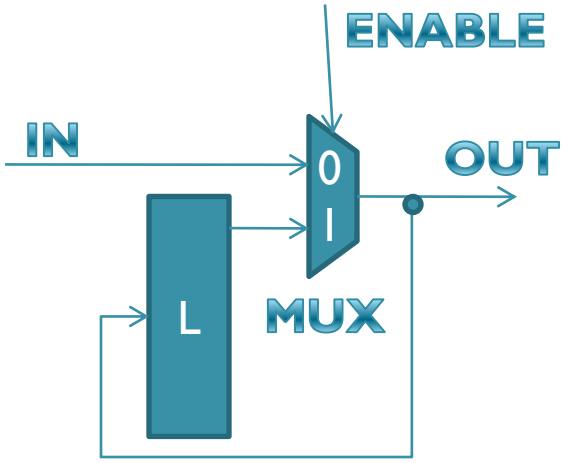


- Supporting variable length of operands.
- Serial input and serial output.
- Auto-configurable cells:
  - The function of a cell depends only on the first digits that pass through;
  - At each step a new cell is configured upon result of the previous cell;

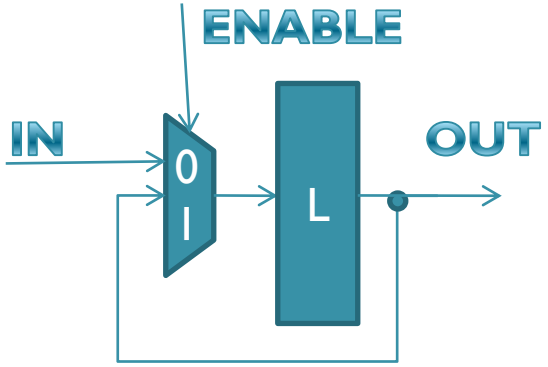
JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

ISI International School for Informatics HAGENBERG

Instant Config vs. Pre-Config



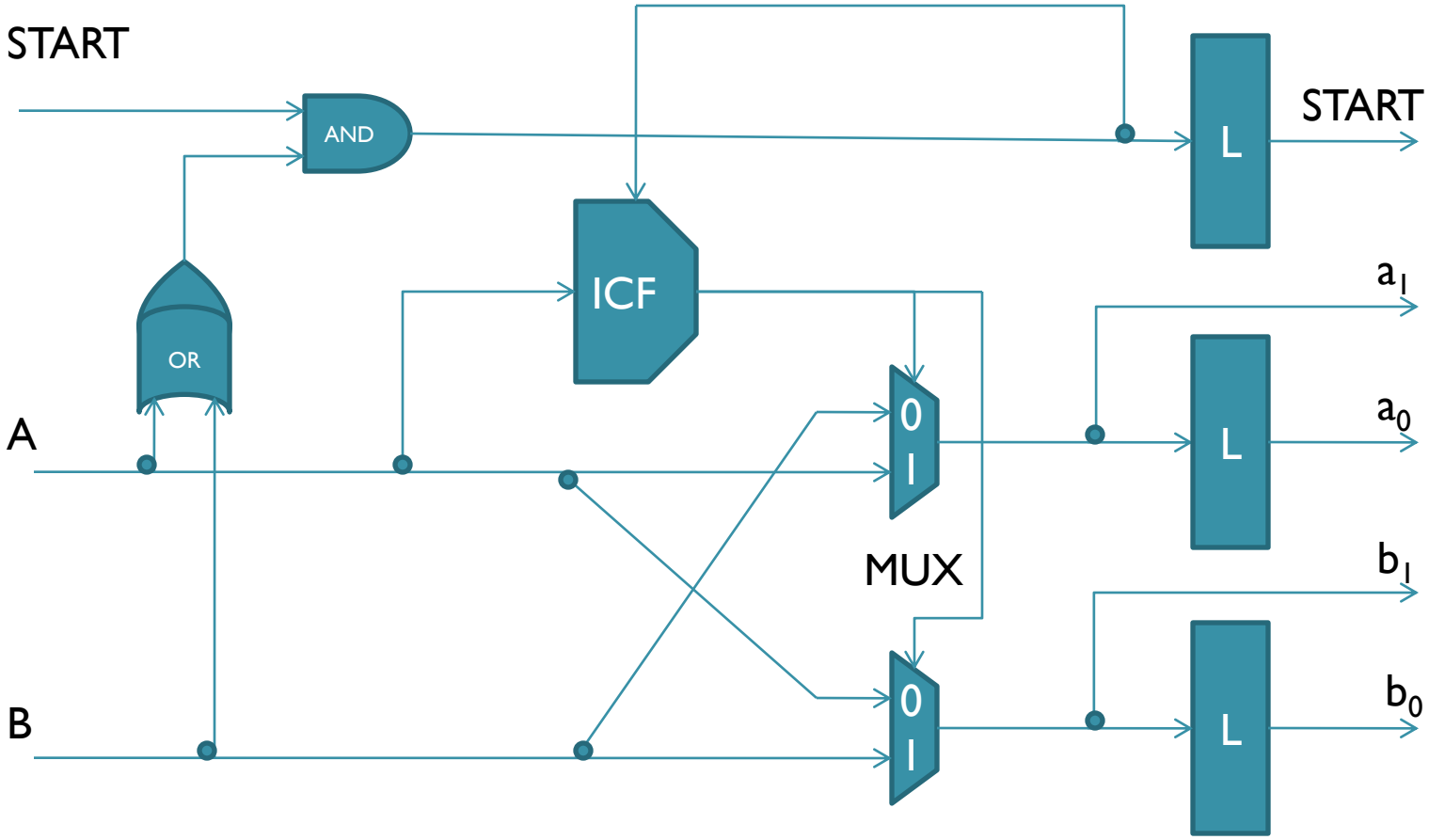
ICF: Instant Config



PCF: Pre-Config

- Simulate auto configurability by multiplexors;
- PCF: when data used for config. *arrives one clock before* data used for comput. (ENABLE signals comput.);
- ICF: when data used for config. *arrives simultaneously* with data used for comput. (ENABLE becomes 1 one clock later);

The Preprocessing Phase



Simulation Overview: NOD(4, 2)

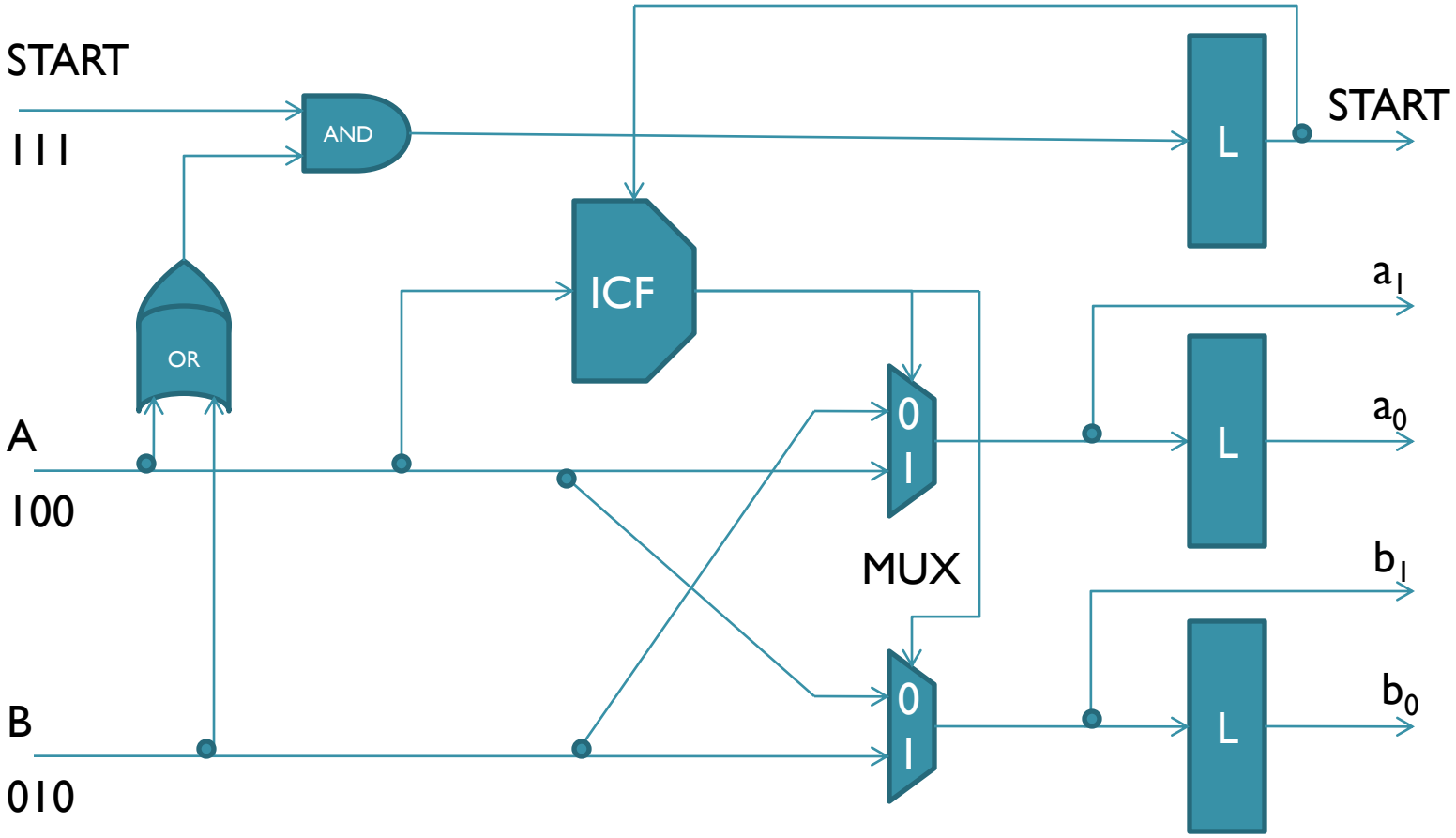
$$\binom{4}{2} \xrightarrow{/2} \binom{2}{1} \xrightarrow{int.} \binom{1}{2} \xrightarrow{b/2} \binom{1}{1} \xrightarrow{-} \begin{pmatrix} 1 \\ \frac{1-1}{4} \end{pmatrix}$$

- The first two steps: preprocessing (shift and interchange) are performed by the feeder P_0.
- The third step (shift b) is performed by P_1.
- The fourth step ((a-b)/4 and interchange) is performed by P_2.
- The result is 1, but due to /2 in the first step we get NOD(4, 2)=2*1=2.

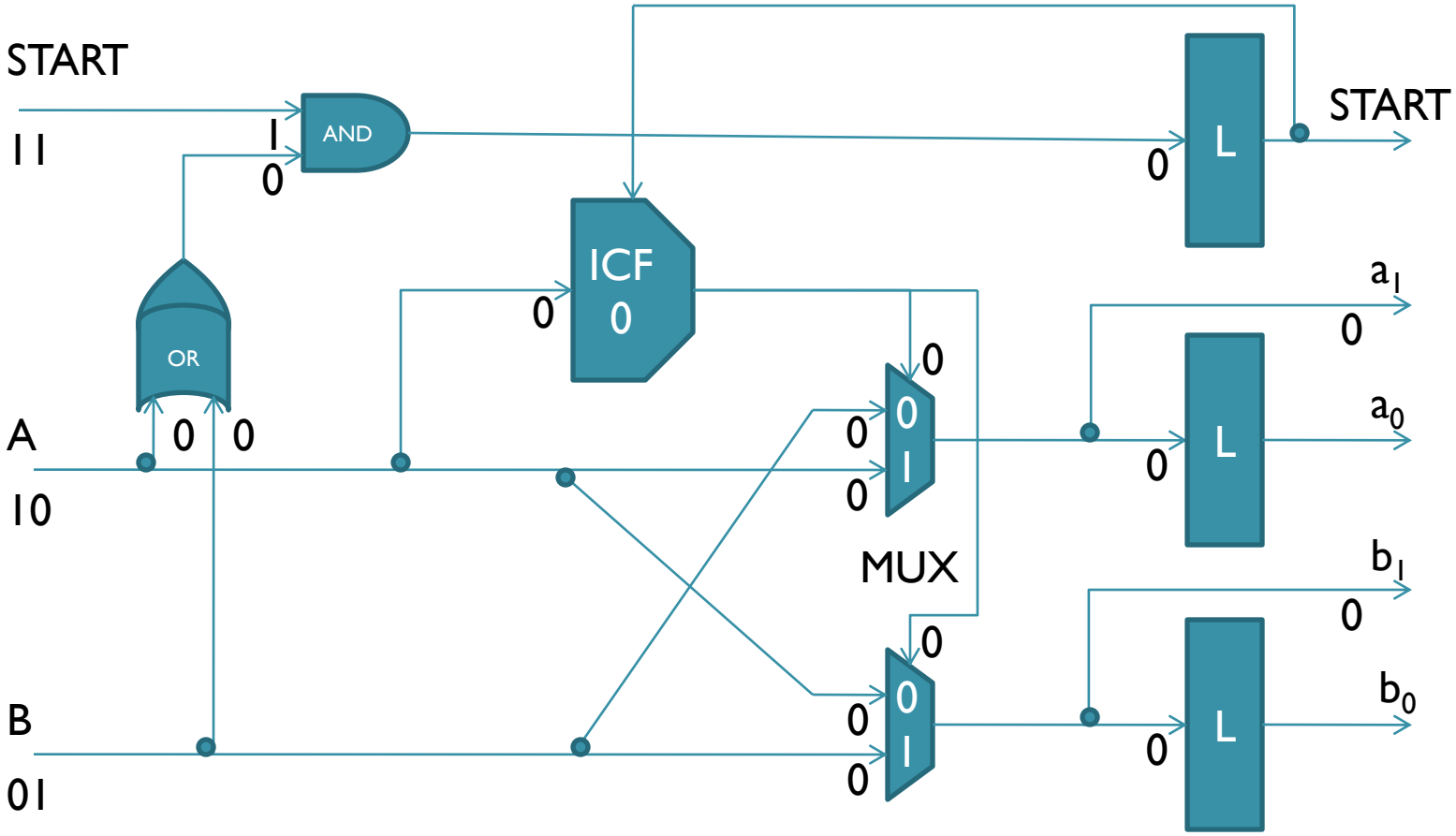
JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

ISI International School for Informatics HAGENBERG

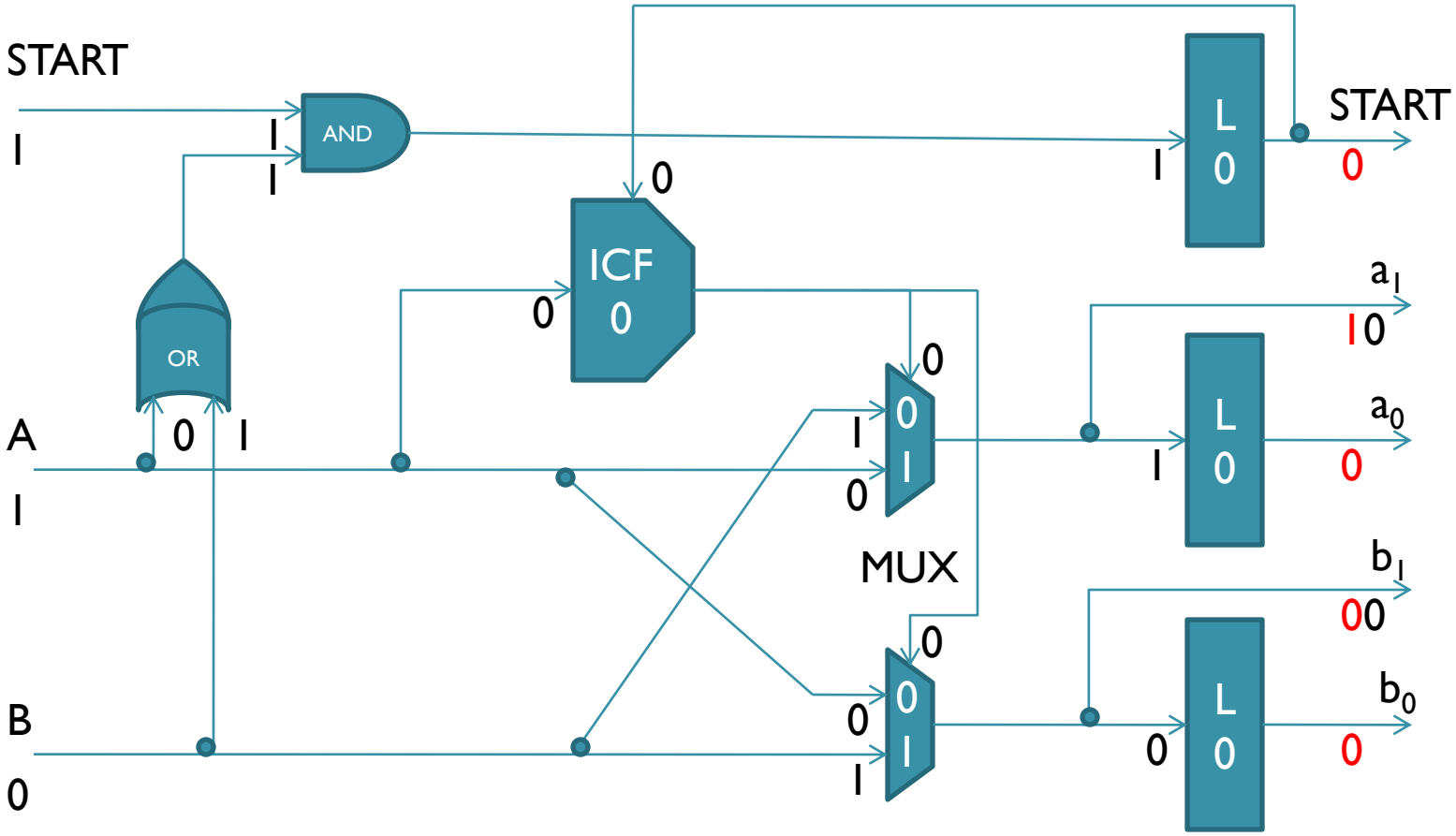
The Preprocessing Phase P_0 (Step 1)



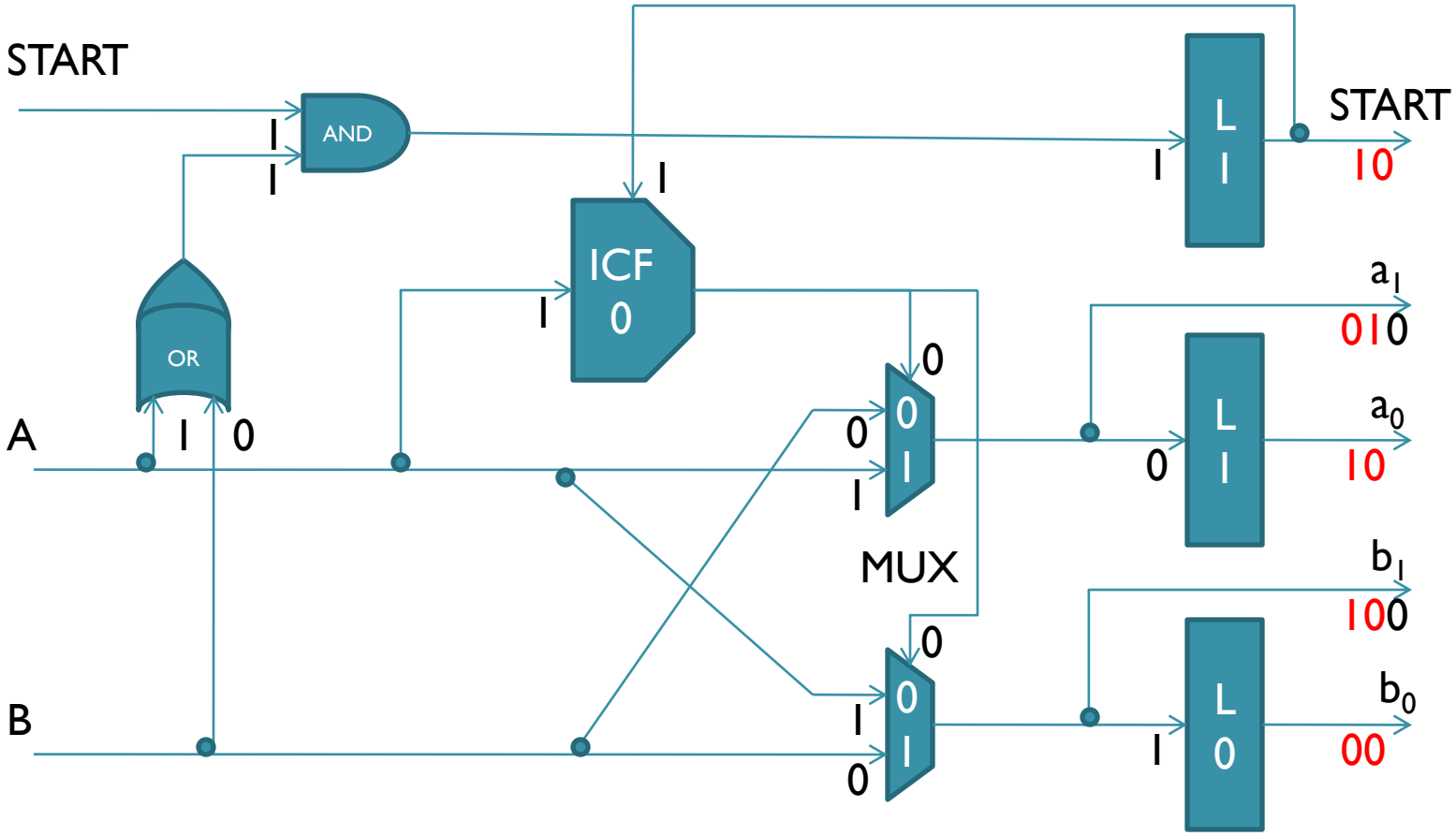
The Preprocessing Phase P_0 (Step 2)



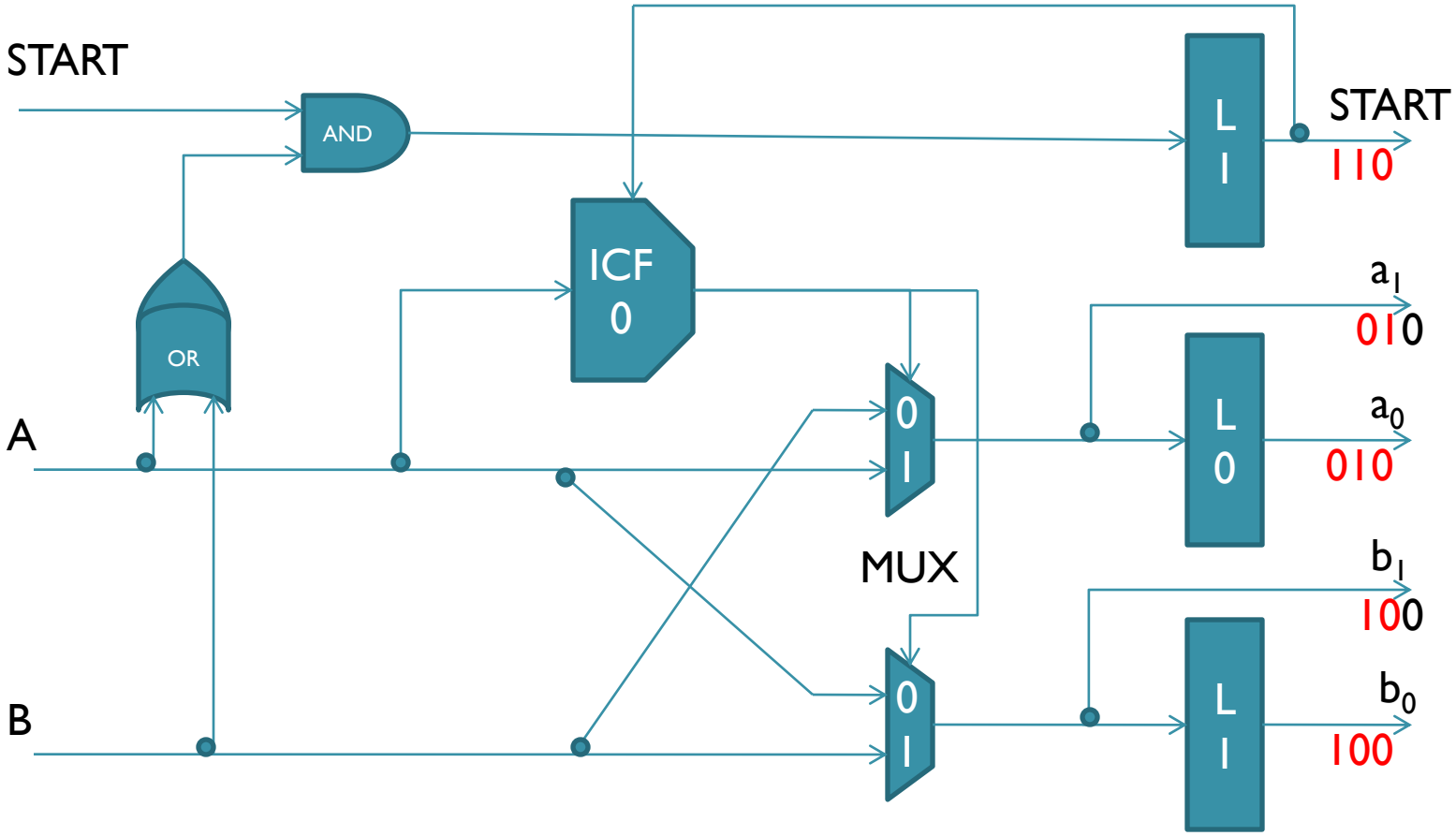
The Preprocessing Phase P_0 (Step 3)



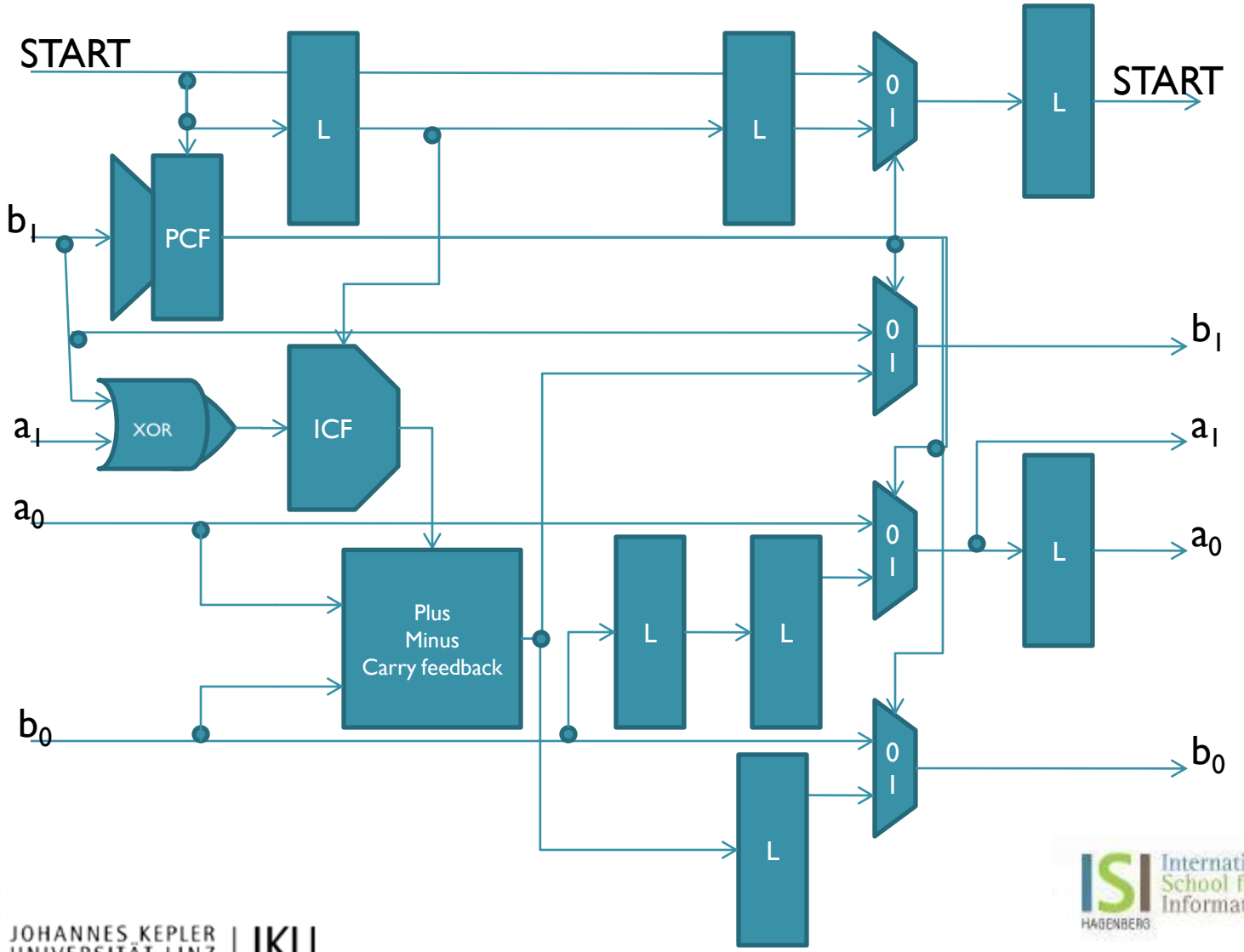
The Preprocessing Phase P_0 (Step 4)



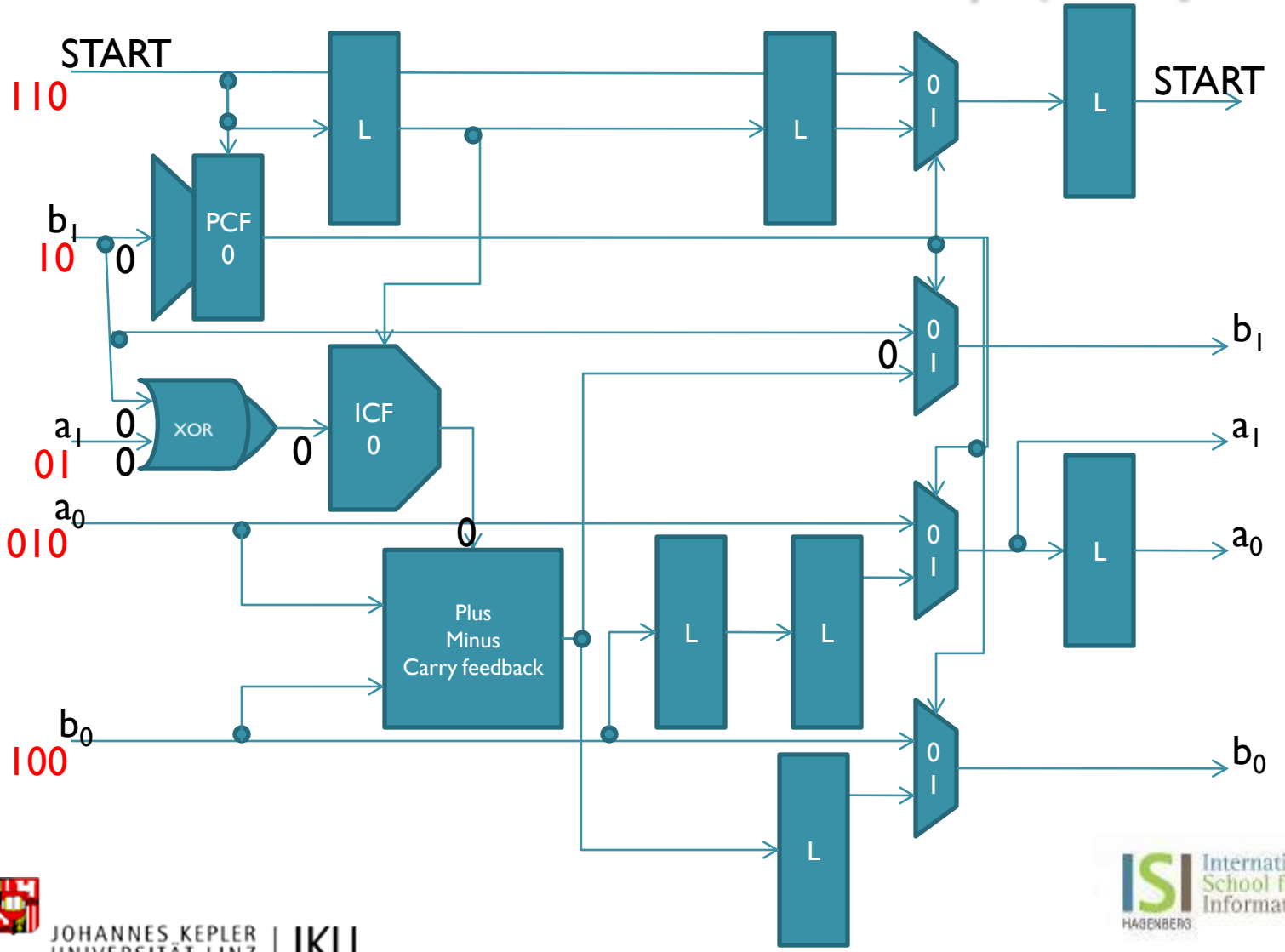
The Preprocessing Phase P_0 (Step 5)



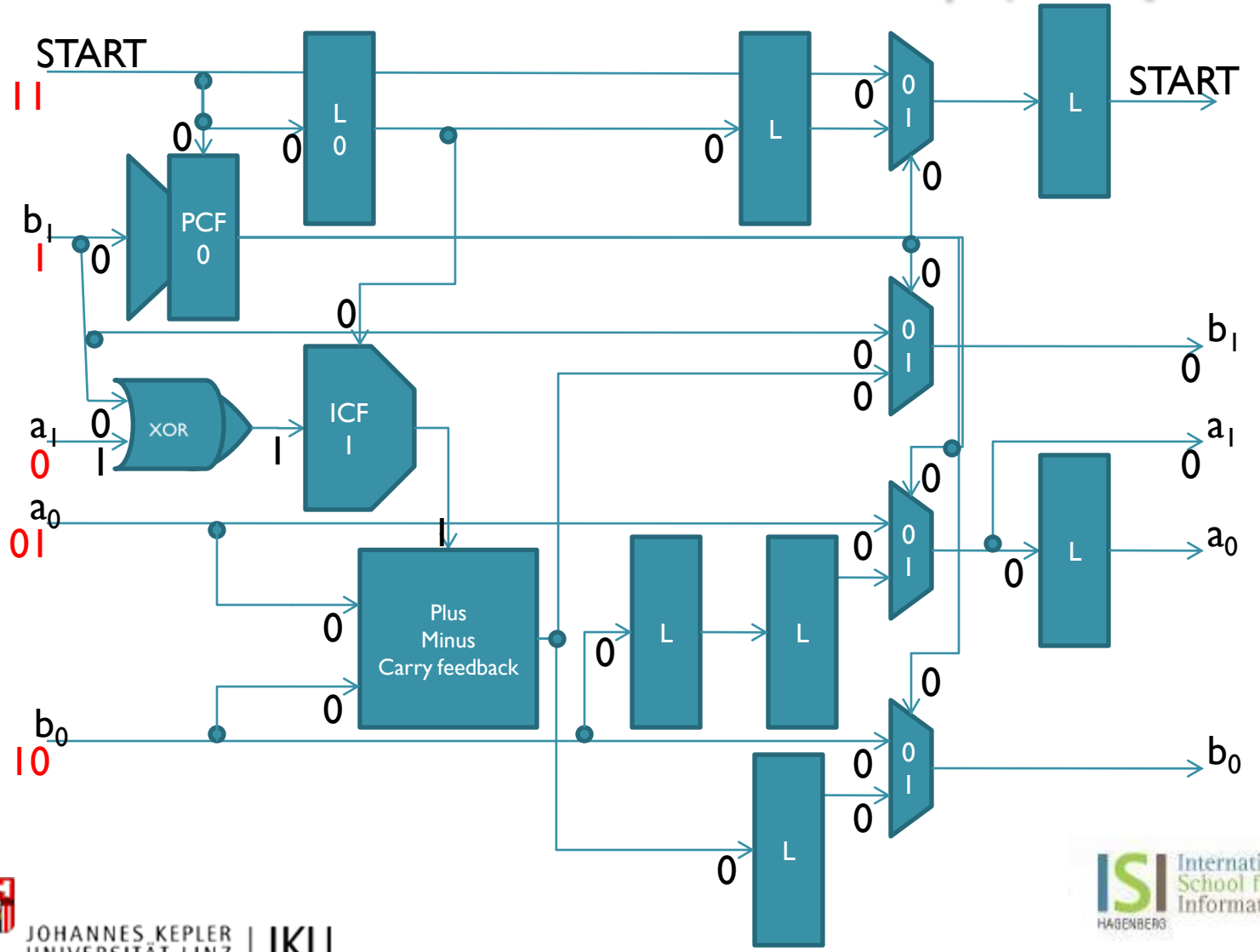
The Reduction Phase



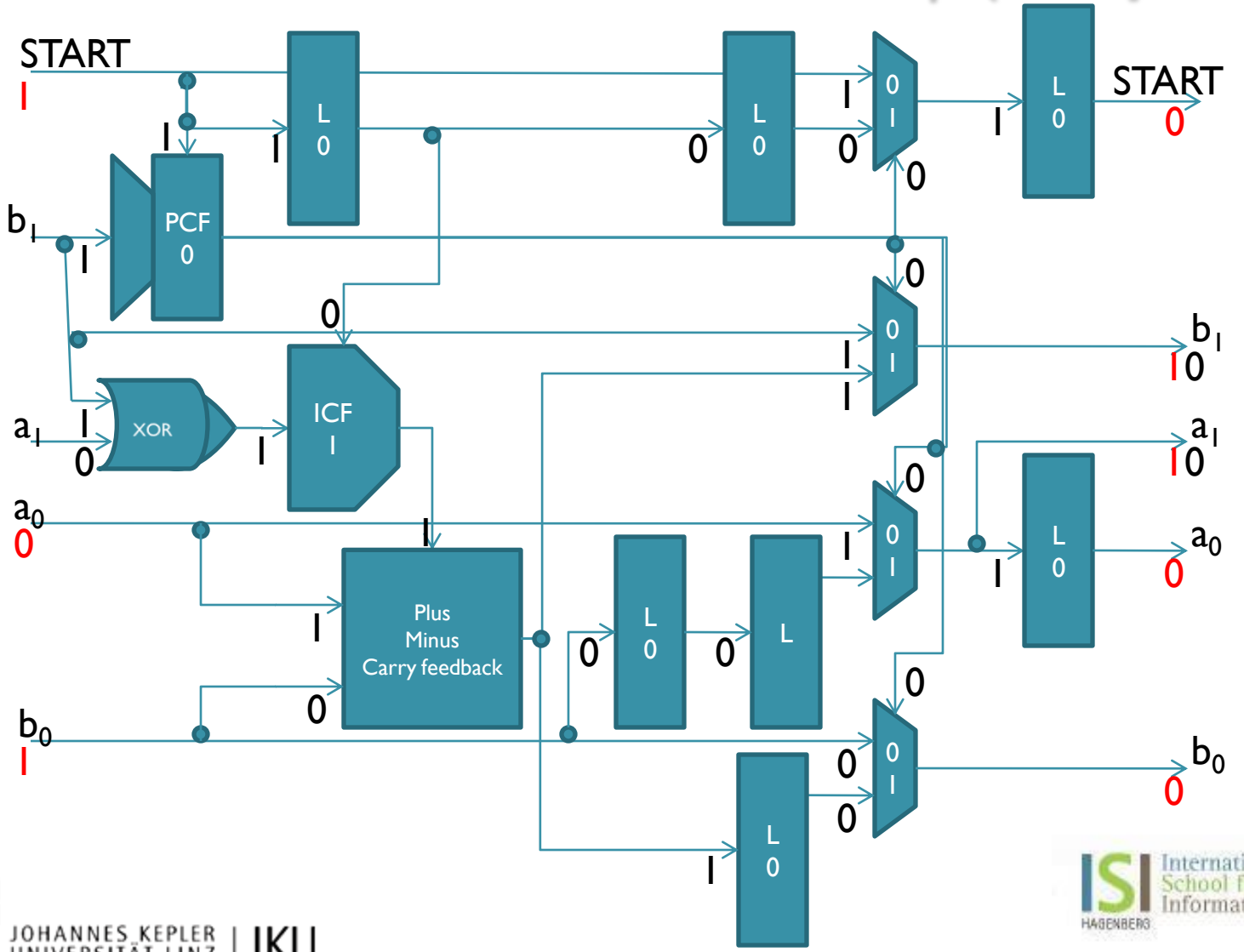
The Reduction Phase P_1 (Step 1)



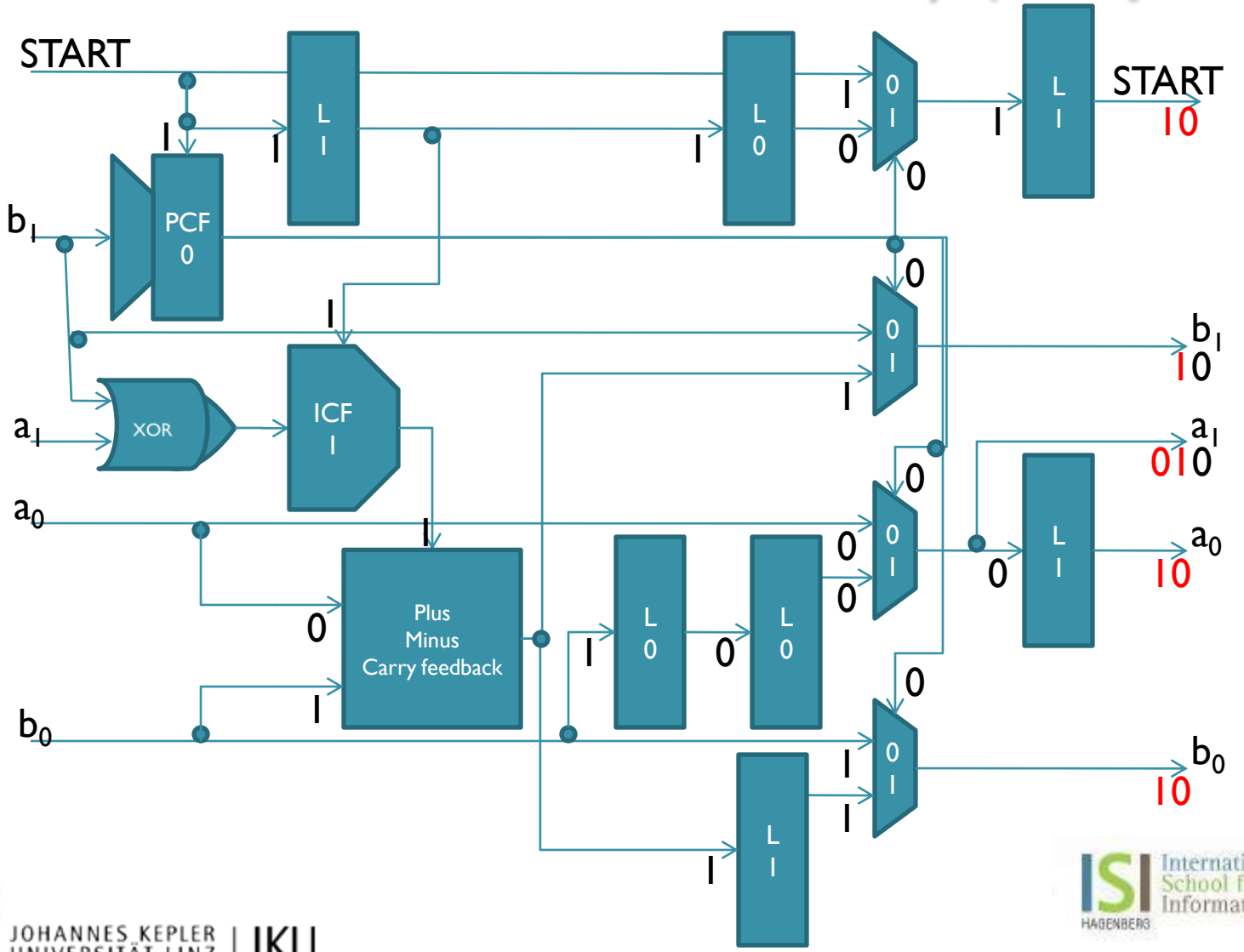
The Reduction Phase P_1 (Step 2)



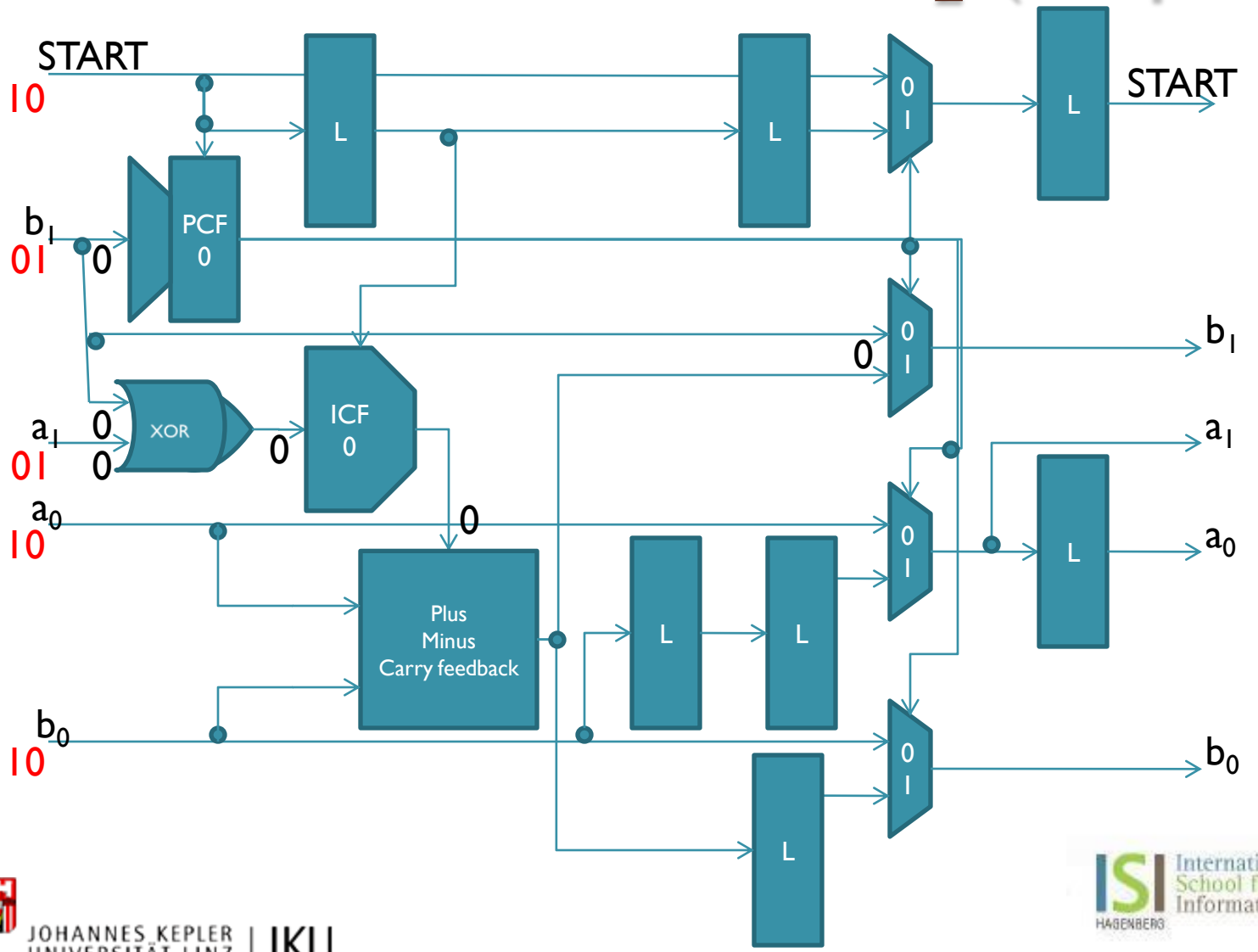
The Reduction Phase P_1 (Step 3)



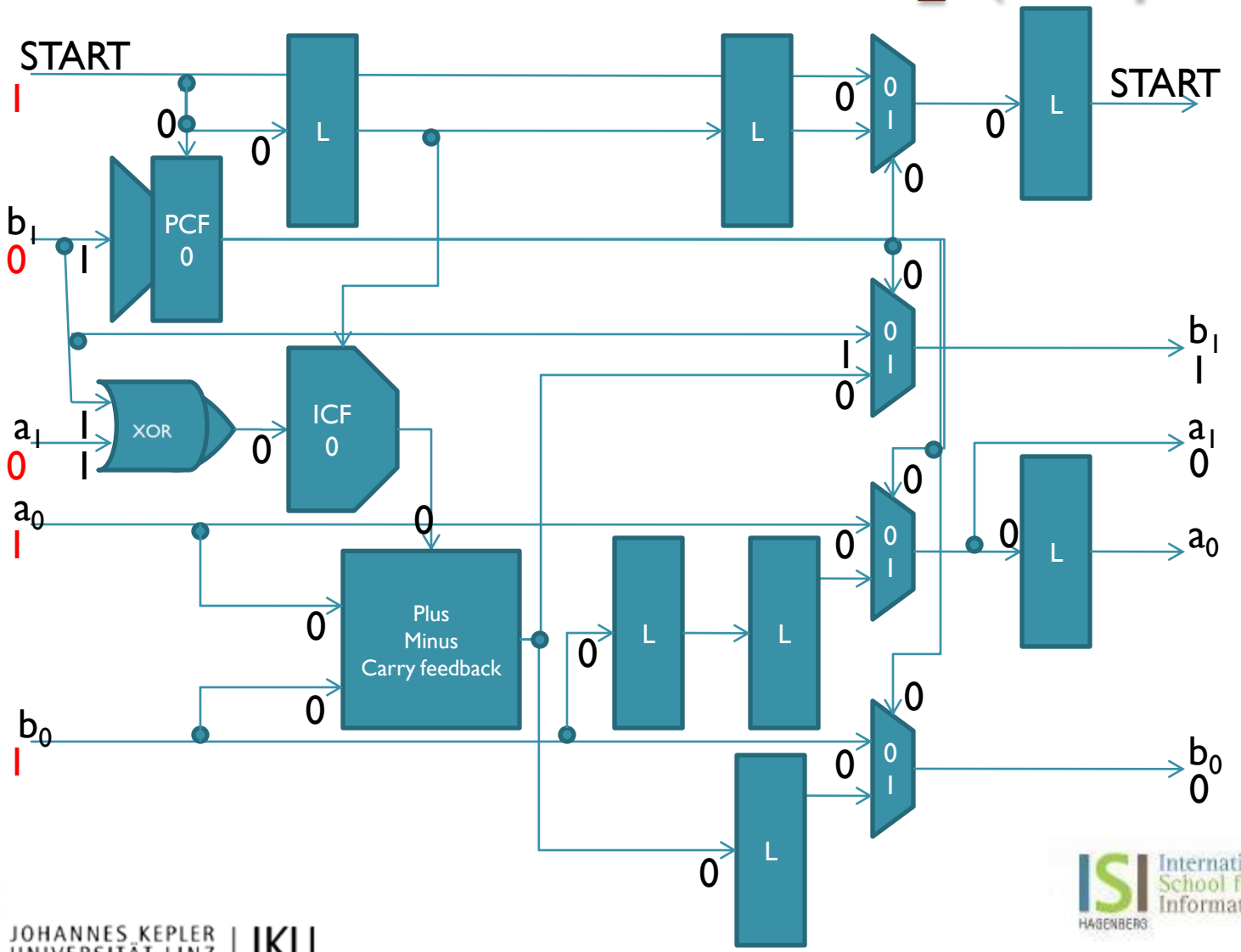
The Reduction Phase P_1 (Step 4)



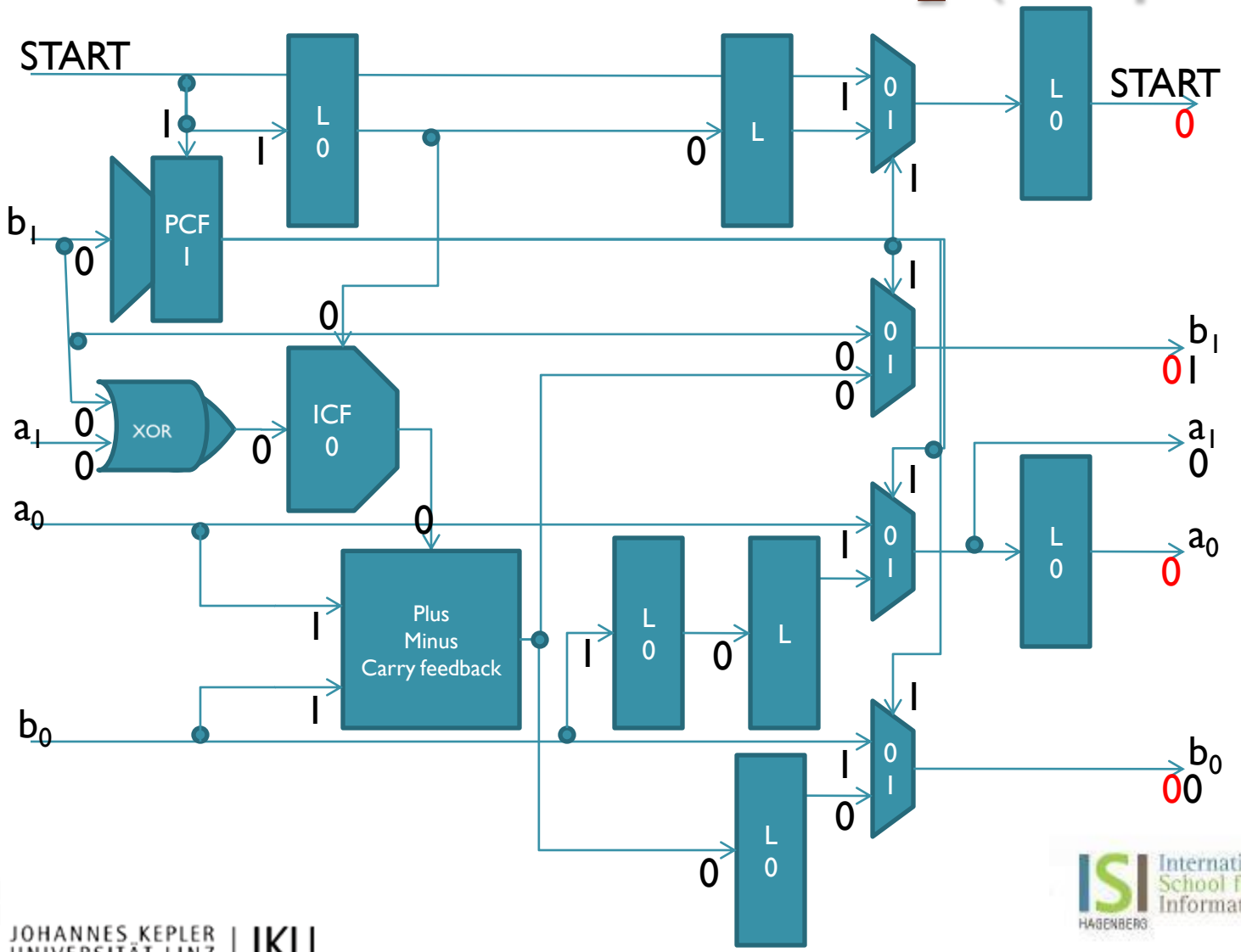
The Reduction Phase P_2 (Step 1)



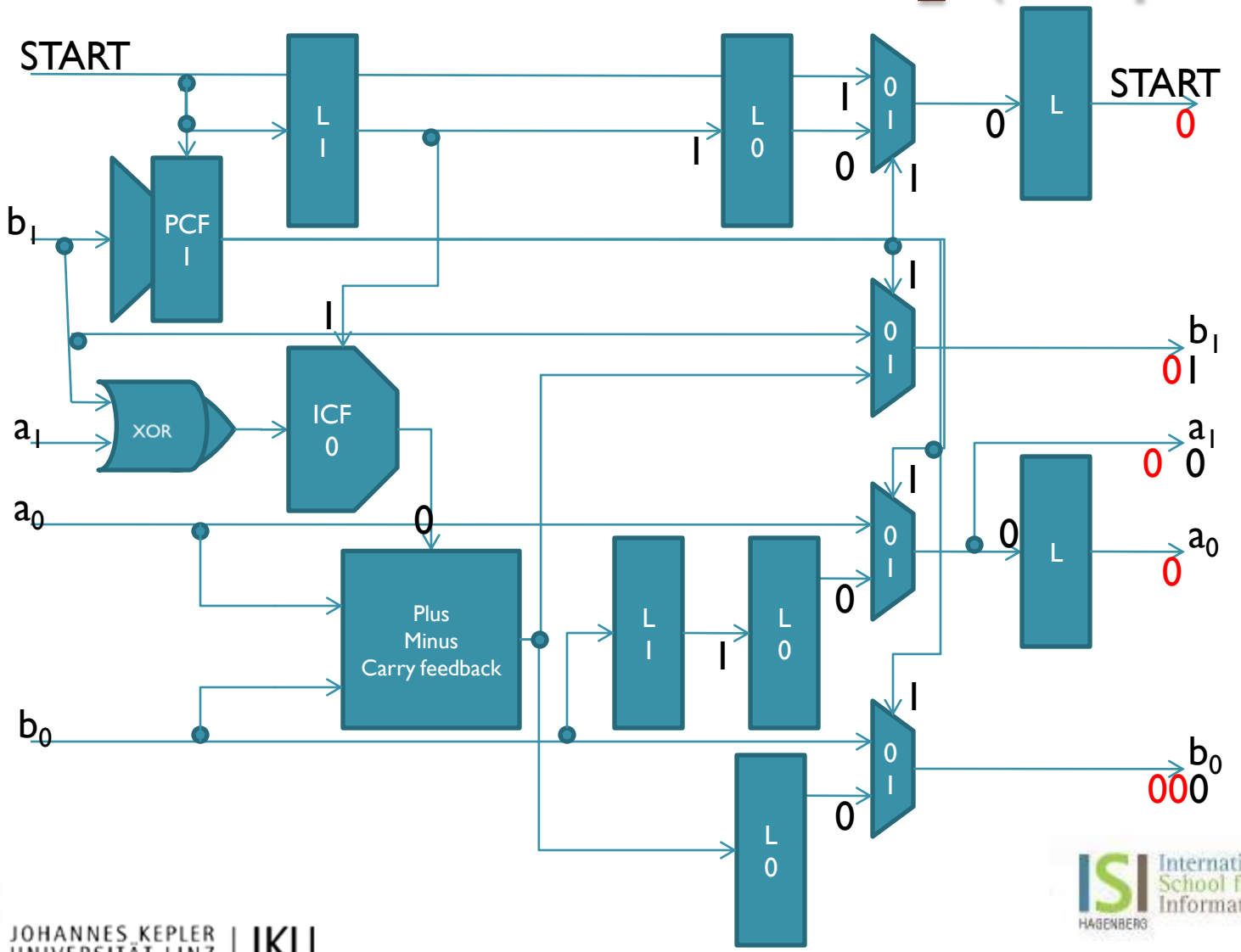
The Reduction Phase P_2 (Step 2)



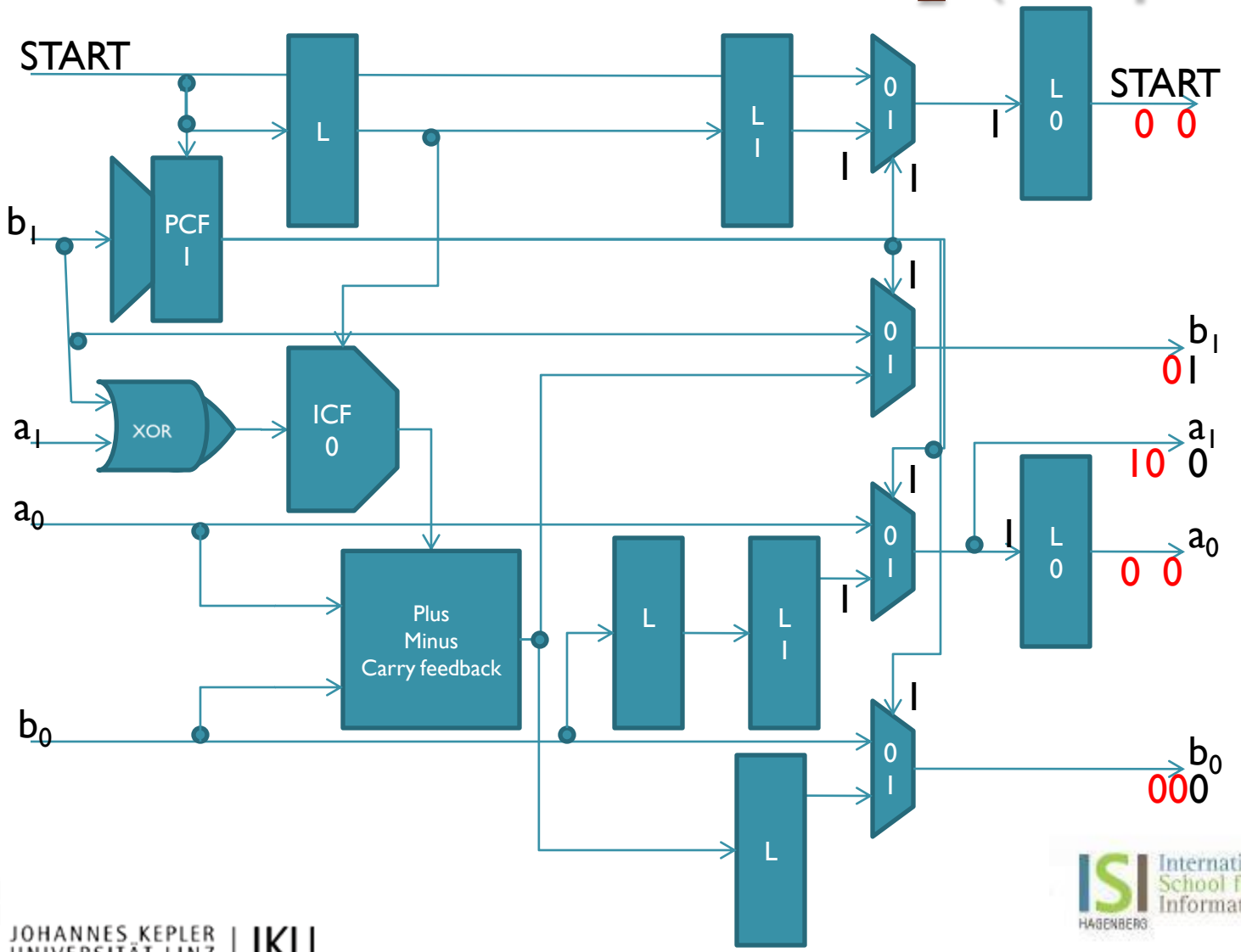
The Reduction Phase P_2 (Step 3)



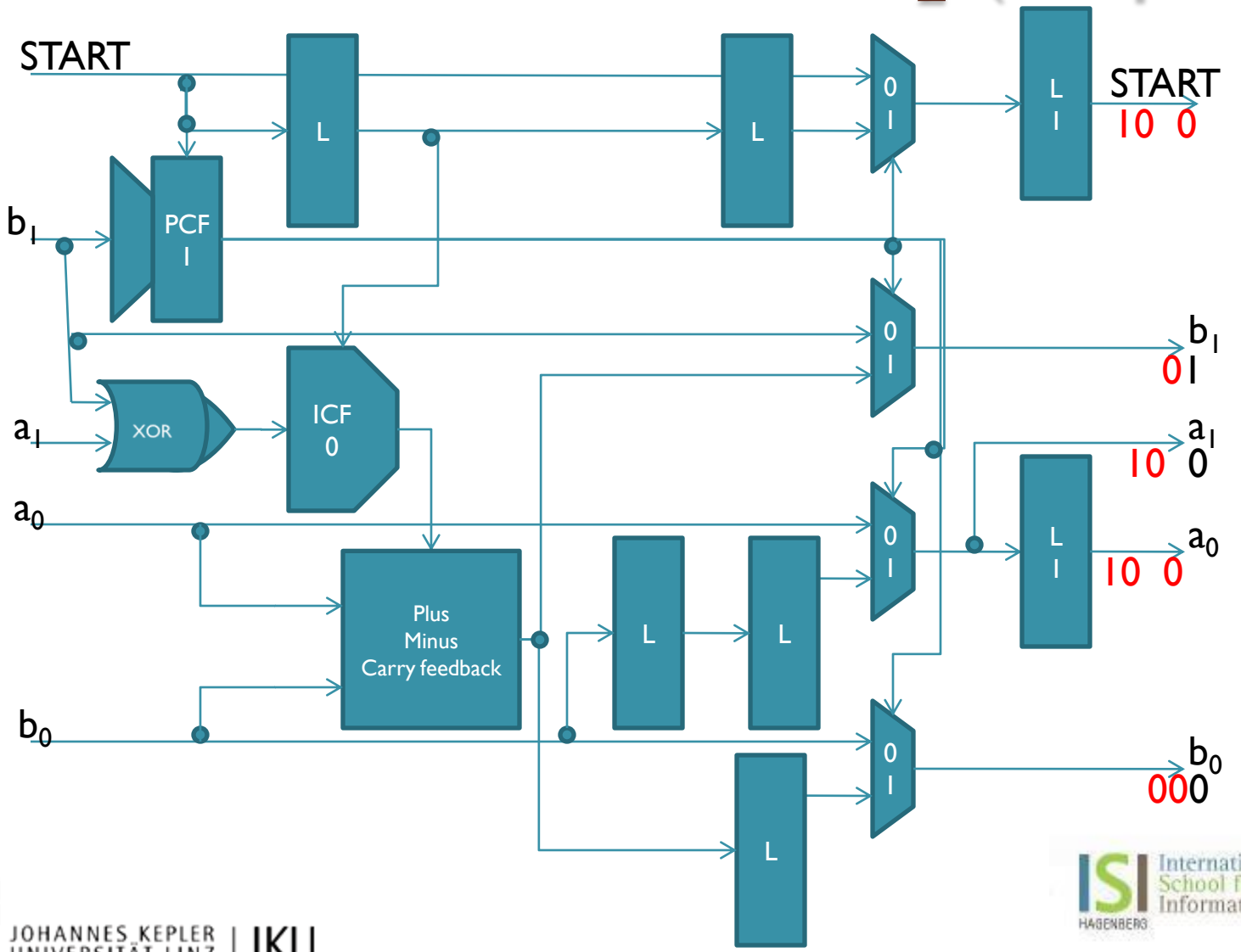
The Reduction Phase P_2 (Step 4)



The Reduction Phase P_2 (Step 5)



The Reduction Phase P_2 (Step 6)



Implementation Notes

- Serial input and serial output
- Termination detection and sign detection
  - not detailed in the presentation;
  - solved by post processing circuit;
- In preprocessing phase:
  - configuration done by ICF (interchange or pass);
  - shifting achieved by leading 0 START signal;
- In reduction phase:
  - Initial configuration done by PCF (shift or +/-)
  - Subsequent configuration done by XOR and ICF

JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

ISI International School for Informatics HAGENBERG

Conclusions

- Systolic design: several cells are placed and routed by simple tiling.
- n-bit operands need in average:
  - 0.75*n shift steps and 0.75*n +/- steps;
  - 1.5*n cells;
  - 3*n clock cycles;
- 100-bit operands can be processed in one pass with a speed-up of 4 over the software solution.
- With longer array speed-up will increase linearly for longer operands
  - linear time complexity of the systolic device;

JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

International School for Informatics HAGENBERG