Cours MPRI "Systèmes synchrones".
Examen final "Circuits synchrones".

All circuits in this problem are *combinatorial* (i.e. memory-free). All circuits will be presented explicitly in two equivalent ways, by :

Schema a *Directed Acyclic Graph* of its structure, properly drawn from inputs to outputs, and documented by the symbolic names from the net-list.

Net-List The *net-list* $v_1 = e_1, \cdots v_n = e_n$ defines each variable v_i by an expression $e_i \in \{t \cap t', t \cup t', t \oplus t'\}$ in which terms t and t' represent : an input ; a constant 0,1 ; a *previously defined* variable, v_j with $j < i$.

The *parity* $p = \mathcal{P}_n(x)$ of a n-bit binary word $x = x_0 \cdots x_{n-1} \in \mathbf{B}^n$ is the word $p \in \mathbf{B}^n$ whose k-th bit is the sum modulo 2 of the previous bits of x :

$$p_k = x_0 \oplus \cdots \oplus x_k = \sum_{j \leq k} x_j \pmod 2. \tag{1}$$

Question 1 (Minimal size Parity circuit)
Describe the minimal size *parity circuit $\mathcal{P}_n$, with n bits of input $x_0 \cdots x_{n-1}$, and n bits of output $p_0 \cdots p_{n-1}$.*

1. *Show that output p_k in your circuit satisfies specification (1).*
2. *Show that it is* the unique minimal size *circuit for computing parity.*
3. *Analyze the combinatorial depth (maximal number of gates between inputs&outputs) of the circuit.*

Question 2 (Minimal depth Parity circuit)

1. *Show that the combinatorial depth q of any parity circuit $\mathcal{P}_n$ is bounded by $q \geq \lceil \log_2(n) \rceil$.*
2. *Construct a* minimal depth *parity circuit $\mathcal{P}_n$, for $n = 2^q$.*
3. *Analyze your circuit : number of gates, depth, wire area.*

A SSA (Single Static Assignement) program is a list of instructions

$$[v_i = e_1, \cdots, v_k = e_k].$$

Each instruction defines integer variable v_j by an expression

$$e_j \in \{t \oplus t', t \cup t', t \cap t', t + t', t - t'\},$$

in which t and t' represent terms which can be, either :

1. an input variable x ;
2. an integer constant ;
3. a previously defined variable v_i, with $i < j$.

For the purpose of this problem, the SSA program is executed on a 16 bits machine. Arithmetic operations are all computed modulo 2^{16}, at the rate of one instruction per cycle : the length k of the SSA code is equal to the program execution time (in cycles).

Question 3 (Optimal Parity software) *1. Provide an SSA program for computing $\mathcal{P}_{16}$. Analyze its length.*

2. *Endow the machine with a 16 bits* shifter. *Add the expression $z^i(t)$ to our SSA instructions. It expresses both up-shift $z^i(t) = 2^i t$ when $i \geq 0$ (expression $t << i$ in C), and down-shift $z^i = t \div 2^{-i}$ when $i < 0$ (expression $t >> -i$ in C). Provide a minimal length SSA program (including shifts) for computing the parity over 16 bits.*
3. *Remove the shifter again, and go back to question 3.1. Is your SSA code for $\mathcal{P}_{16}$ of minimal length ? If so, prove it ; else, provide a counter example with (say) less than 30 SSA instructions (without shift).*