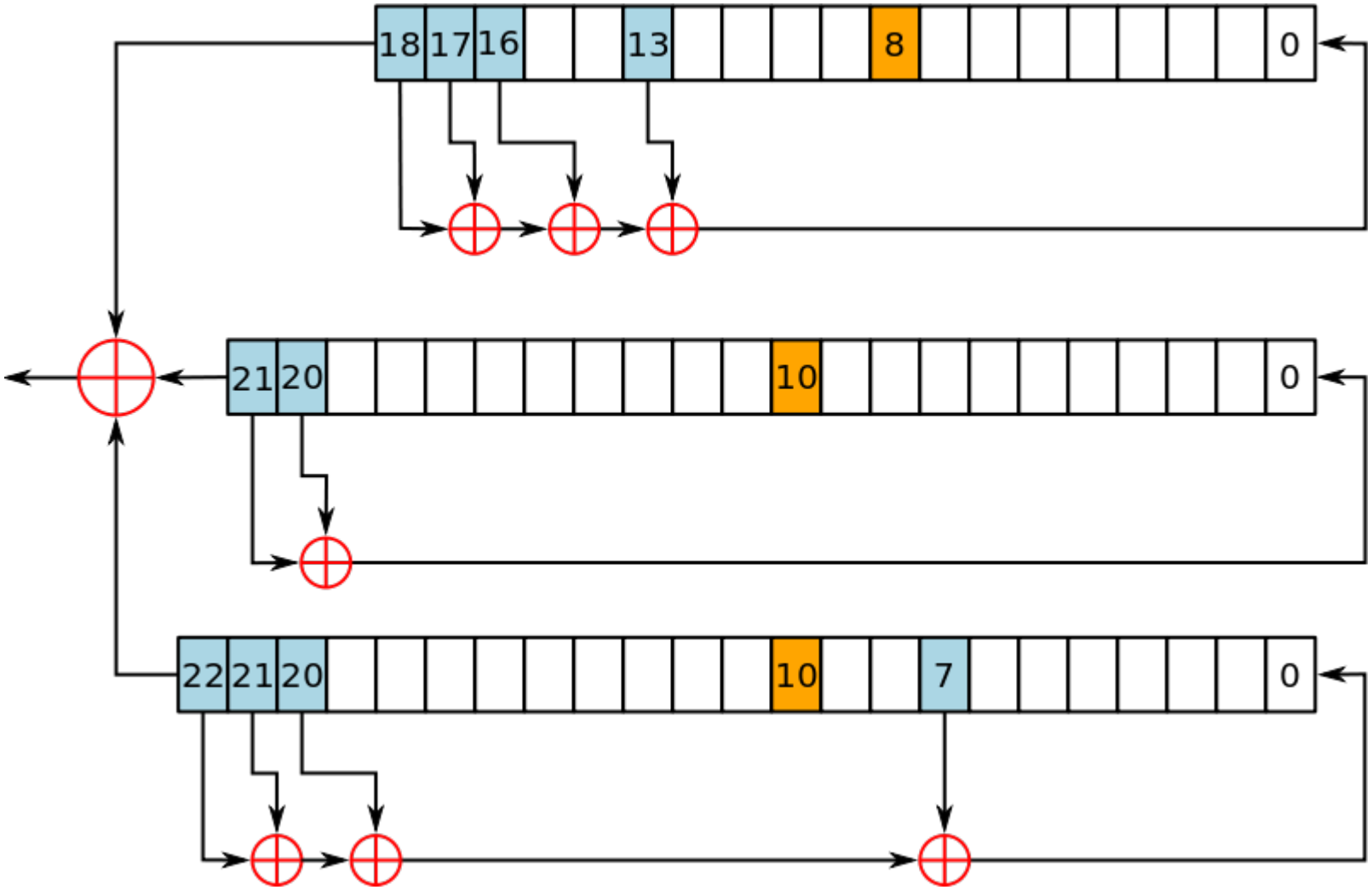
Stream Cipher Systems
GSM Ciphers

A5/1 GSM Cipher:



LFSR number	Length in bits	Feedback polynomial	Clocking bit	Tapped bits
1	19	$x^{19} + x^{18} + x^{17} + x^{14} + 1$	8	13, 16, 17, 18
2	22	$x^{22} + x^{21} + 1$	10	20, 21
3	23	$x^{23} + x^{22} + x^{21} + x^{8} + 1$	10	7, 20, 21, 22

The bits are indexed with the least significant bit (LSB) as 0.

$$X = Maj(R_1[8], R_2[10], R_3[10])$$

$$R_1 \ is \ \ clocked \ if \ \ X = R_1[8]$$

$$R_2 \ is \ \ clocked \ if \ \ X = R_2[10]$$

$$R_3 \ is \ \ clocked \ if \ \ X = R_3[10]$$

Majority Function:

$$Maj(x, y, z) = xy \oplus yz \oplus zx$$

A5/2 GSM Cipher:

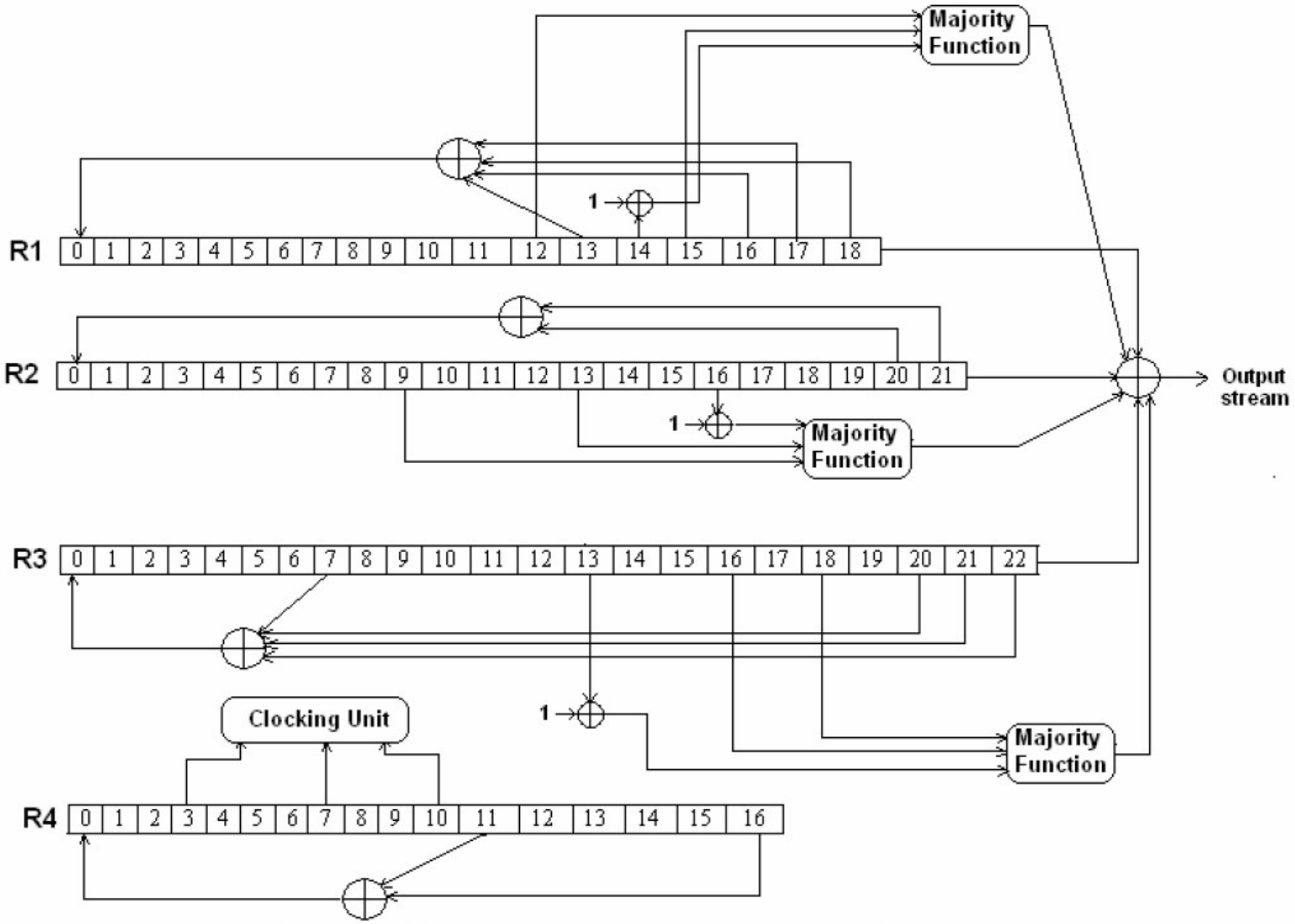


Figure 2 - The A5/2 Stream Cipher

$$Z = X_1 \oplus X_2 \oplus X_3 \oplus R_1[18] \oplus R_2[21] \oplus R_3[22]$$

$$X_1 = Maj(R_1[12], 1 \oplus R_1[14], R_1[15])$$

$$X_2 = Maj(R_2[9], R_2[13], 1 \oplus R_2[16])$$

$$X_3 = Maj(1 \oplus R_3[13], R_3[16], R_3[18])$$

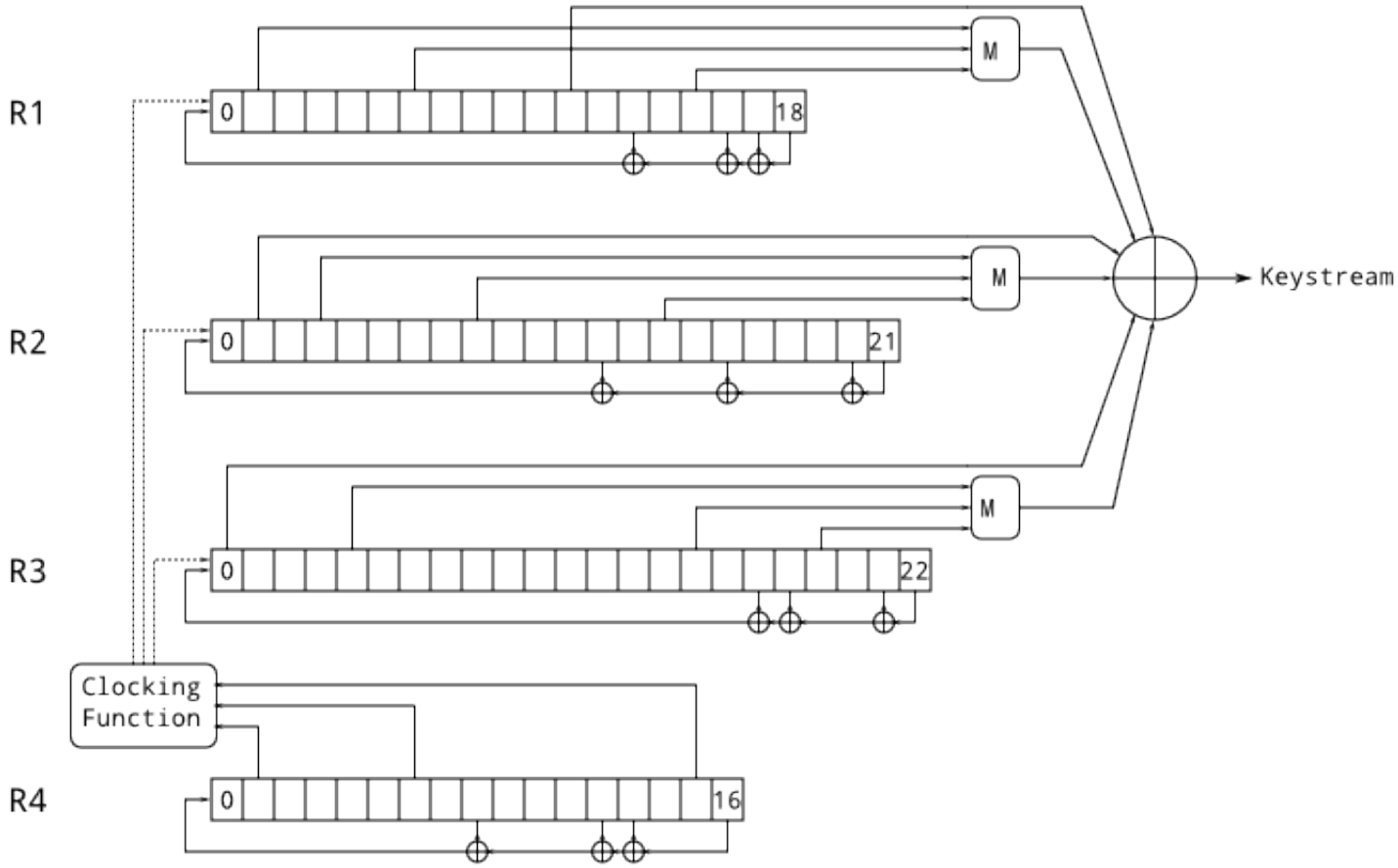
$$X_4 = Maj(R_4[3], R_4[7], R_4[10])$$

$$R_1 \ is \ \ clocked \ if \ \ X_4 = R_4[10]$$

$$R_2 \ is \ \ clocked \ if \ \ X_4 = R_4[3]$$

$$R_3 \ is \ \ clocked \ if \ \ X_4 = R_4[7]$$

A5-GMR-1



A5-GMR-1

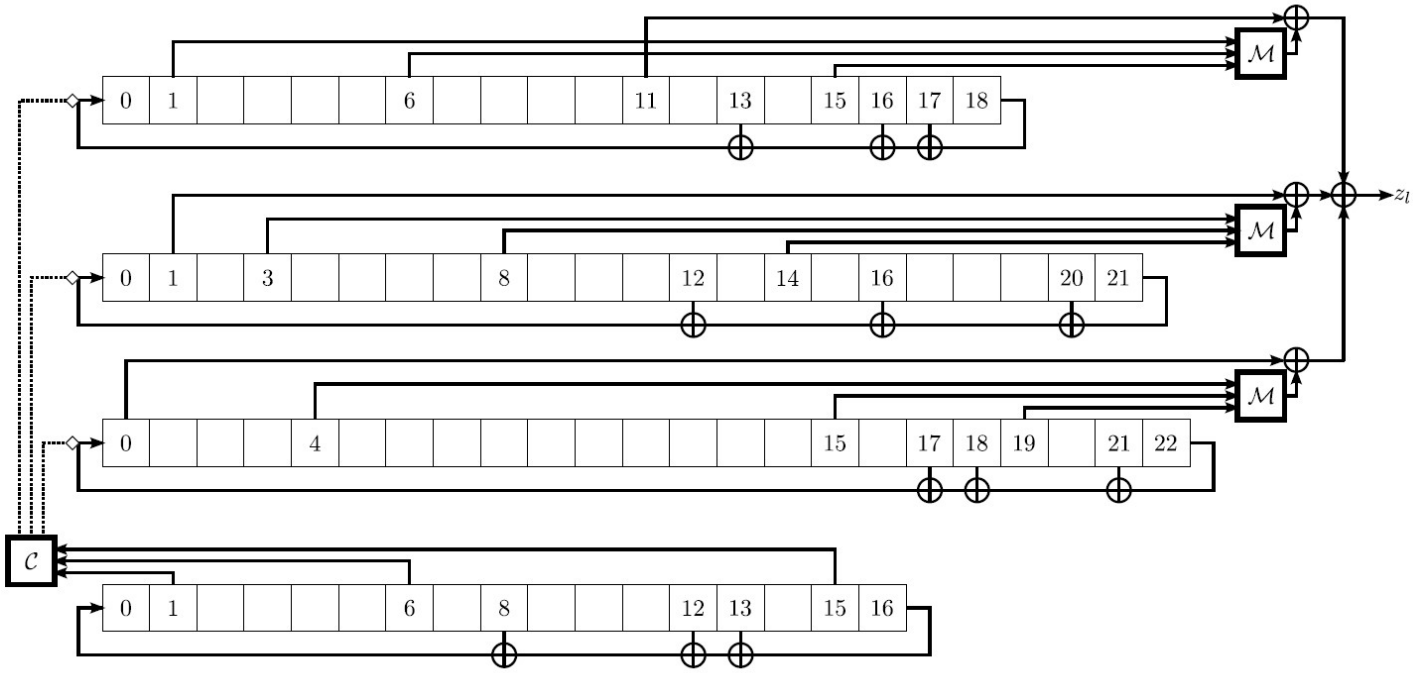


Figure 2: The A5-GMR-1 cipher

Trivium
Stream Cipher

Trivium is a relatively new stream cipher which uses an 80-bit key. It is based on a combination of three shift registers. Even though these are feedback shift registers, there are nonlinear components used to derive the output of each register, unlike the LFSRs that we studied in the previous section.

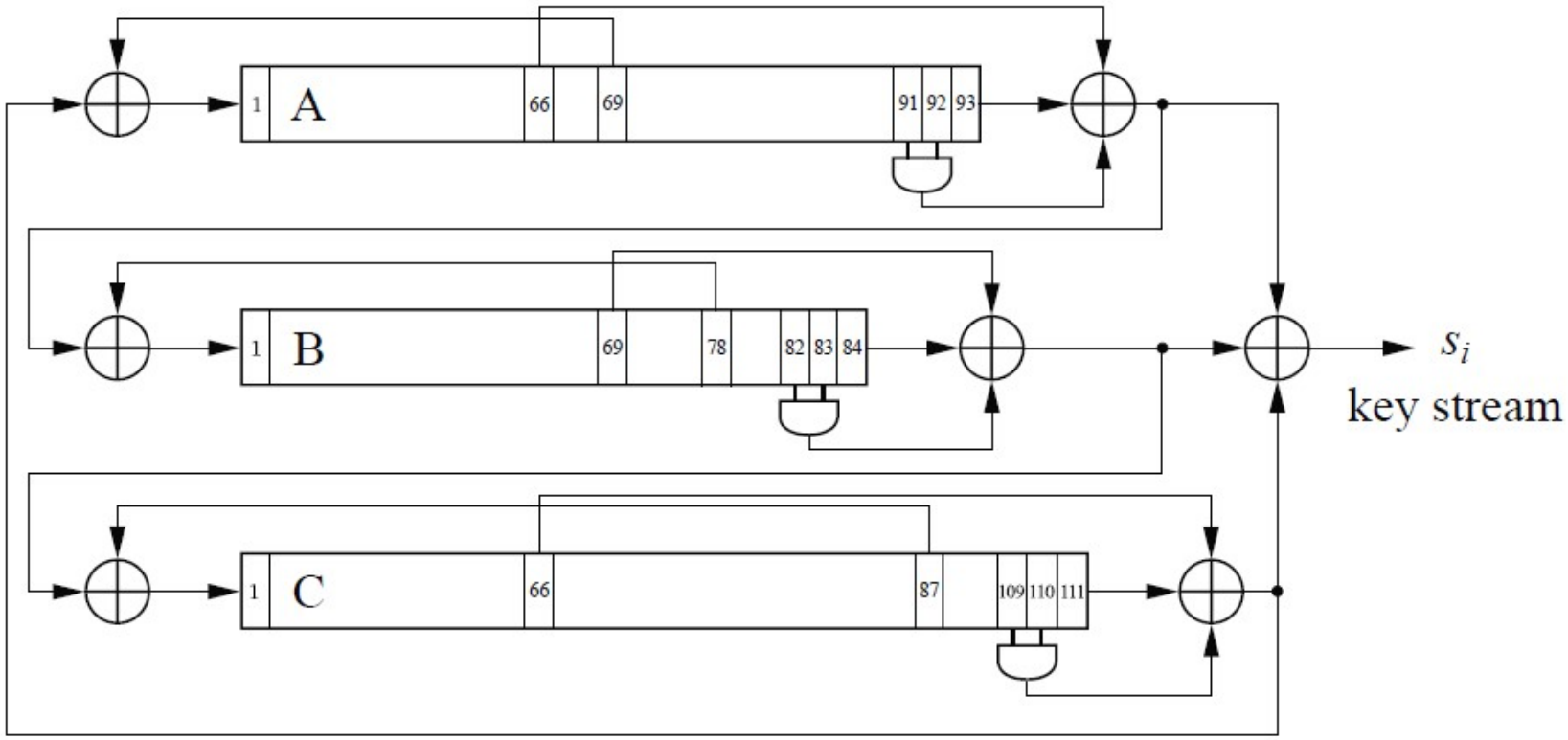
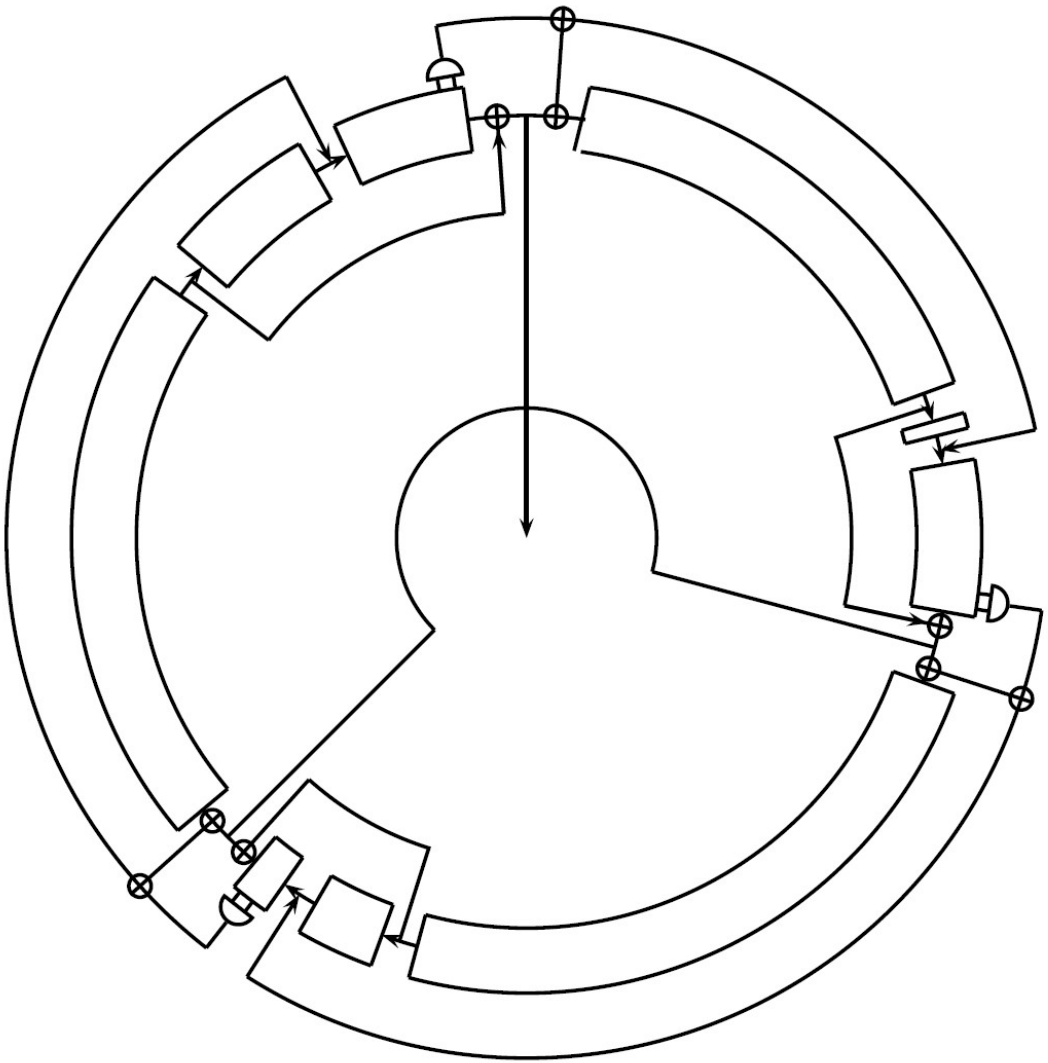


Fig. 2.8 Internal structure of the stream cipher Trivium



the output of each register is connected to the input of another register. Thus, the registers are arranged in circle-like fashion. The cipher can be viewed as consisting of one circular register with a total length of $93 + 84 + 111 = 288$. Each of the three registers has similar structure as described below.

The input of each register is computed as the XOR-sum of two bits:

- The output bit of another register according to Fig. 2.8. For instance, the output of register *A* is part of the input of register *B*.
- One register bit at a specific location is fed back to the input. The positions are given in Table 2.4. For instance, bit 68 of register *A* is fed back to its input.

The output of each register is computed as the XOR-sum of three bits:

- The rightmost register bit.
- One register bit at a specific location is fed forward to the output. The positions are given in Table 2.4. For instance, bit 66 of register *A* is fed to its output.
- The output of a logical AND function whose input is two specific register bits. Again, the positions of the AND gate inputs are given in Table 2.4.

Specification of Trivium

	register length	feedback bit	feedforward bit	AND inputs
A	93	69	66	91, 92
B	84	78	69	82, 83
C	111	87	66	109, 110

Encryption with Trivium

Almost all modern stream ciphers have two input parameters: a key k and an initialization vector IV. The former is the regular key that is used in every symmetric crypto system. The IV serves as a randomizer and should take a new value for every encryption session. It is important to note that the IV does not have to be kept secret, it merely must change for every session. Such values are often referred to as *nonces*, which stands for "number used once". Its main purpose is that two key streams produced by the cipher should be different, even though the key has not changed. If this were not the case, the following attack becomes possible. If an attacker has known plaintext from a first encryption, he can compute the corresponding key stream. The second encryption using the same key stream can now immediately be deciphered. Without a changing IV, stream cipher encryption is highly deterministic. Methods for generating IVs are discussed in Sect. 5.1.2. Let's look at the details of running Trivium:

Initialization Initially, an 80-bit IV is loaded into the 80 leftmost locations of register A, and an 80-bit key is loaded in the 80 leftmost locations of register B. All other register bits are set to zero with the exception of the three rightmost bits of register C, i.e., bits c_{109}, c_{110} and c_{111}, which are set to 1.

Warm-up Phase In the first phase, the cipher is clocked $4 \times 288 = 1152$ times. No cipher output is generated.

Encryption Phase The bits produced hereafter, i.e., starting with the output bit of cycle 1153, form the key stream.

The warm-up phase is needed for randomizing the cipher sufficiently. It makes sure that the key stream depends on both the key k and the IV.

An attractive feature of Trivium is its compactness, especially if implemented in hardware. It mainly consists of a 288-bit shift register and a few Boolean operations. It is estimated that a hardware implementation of the cipher occupies and area of between about 3500 and 5500 gate equivalences, depending on the degree of parallelization. (A gate equivalence is the chip area occupied by a 2-input NAND gate.) For instance, an implementation with 4000 gates computes the key stream at a rate of 16 bits/clock cycle. This is considerably smaller than most block ciphers such as AES and is very fast. If we assume that this hardware design is clocked at a moderate 125 MHz, the encryption rate would be $16\text{bit} \times 125\text{MHz} = 2$ Gbit/sec. In software, it is estimated that computing 8 output bits takes 12 cycles on a 1.5 GHz Intel CPU, resulting in a theoretical encryption rate of 1 Gbit/sec.