Research Inventy: International Journal of Engineering And Science
Vol.15 , Issue 9 (September 2025), PP -21-26
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

RESEARCH ON DECODING THE DATA TRANSMISSION FROM AIRCRAFT CARRYING TO SEMI-ACTIVE RADIO-GUIDED MISSILES

Dang Vo Cong, Hoang Van Tam, Hoang Huy Le, Nguyen Trong Hieu, Air Defense-Air Force Technical Institute, Ha Noi, Viet Nam.
Corresponding Author: mrcong.vkt@gmail.com

Abstract: *This paper presents an algorithm for decoding command codes in the transmission line from new generation aircraft (Su-27, Su-30) to the several types of radio-guided missiles (for example AA-10, AA-12 missiles). The article addresses three main contents: first, presenting the characteristics of signals containing command codes from new generation aircraft sent down to control the several types of radio- guided missiles; second, presenting the algorithm for decoding command codes in signals transmitted from new generation aircraft down to the several types of radio-guided missiles. The algorithm is applied to design hardware on FPGA technology. In the paper, experimental results were carried out with AA-10 missile.*

Keywords: *Aircraft, Missile, Bipolar code.*

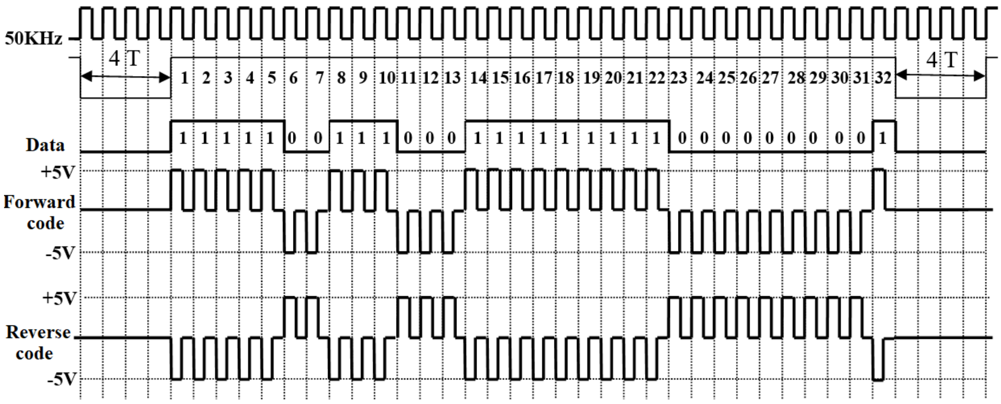
Date Of Submission: 08-09-2025 Date Of Acceptance: 19-09-2025

I. INTRODUCTION

For older generation missiles such as AA-2C, AA-8, AS-10, AS-14, ... two-way communication between new generation aircraft and missiles is conducted through single-digit commands or analog signals. This means when missile control is needed, the weapon control system of the new generation aircraft (WCS) will transmit commands through digital signals (present/absent) in the form of 27V DC voltage or transmit target parameters through analog signals.

However, for new generation missiles in general and the radio-guided missiles in particular, communication between new generation aircraft and missiles has shifted to using serial transmission lines carrying information in the form of bipolar codes. Specifically, command code transmission from Su-30 aircraft to radio-guided missiles uses two forward and reverse transmission lines, with information encoded in 32-bit bipolar code format. The 32-bit bipolar code is commonly used in military technology, especially in aviation. The general diagram of the 32-bit bipolar code sequence used in communication between Su-27, Su-30 aircraft and radio-guided missiles is shown in Figure 1.

Figure 1. Characteristics of 32-bit serial bipolar code



Before missile launch, through a two-wire bipolar communication link (1 wire carrying forward code, 1 wire carrying reverse code), 9 analog parameters and 13 command parameters are transmitted to the seeker head of the radio-guided missiles from the WCS system of the new generation aircraft [1]. Information about command codes is transmitted continuously by code words, each code word is 32 bits long with a cycle period of T=20µs for each bit, corresponding to a frequency of 50Hz, with spacing between code words of 4T=80µs [1].

For forward code signals: logic level 0 corresponds to a pulse with the first half-cycle at +5V voltage, second half-cycle at 0V voltage. For reverse code, it is opposite: logic level 0 corresponds to a pulse with the first half-cycle at -5V, second half-cycle at 0V.

II. ALGORITHM DEVELOPMENT

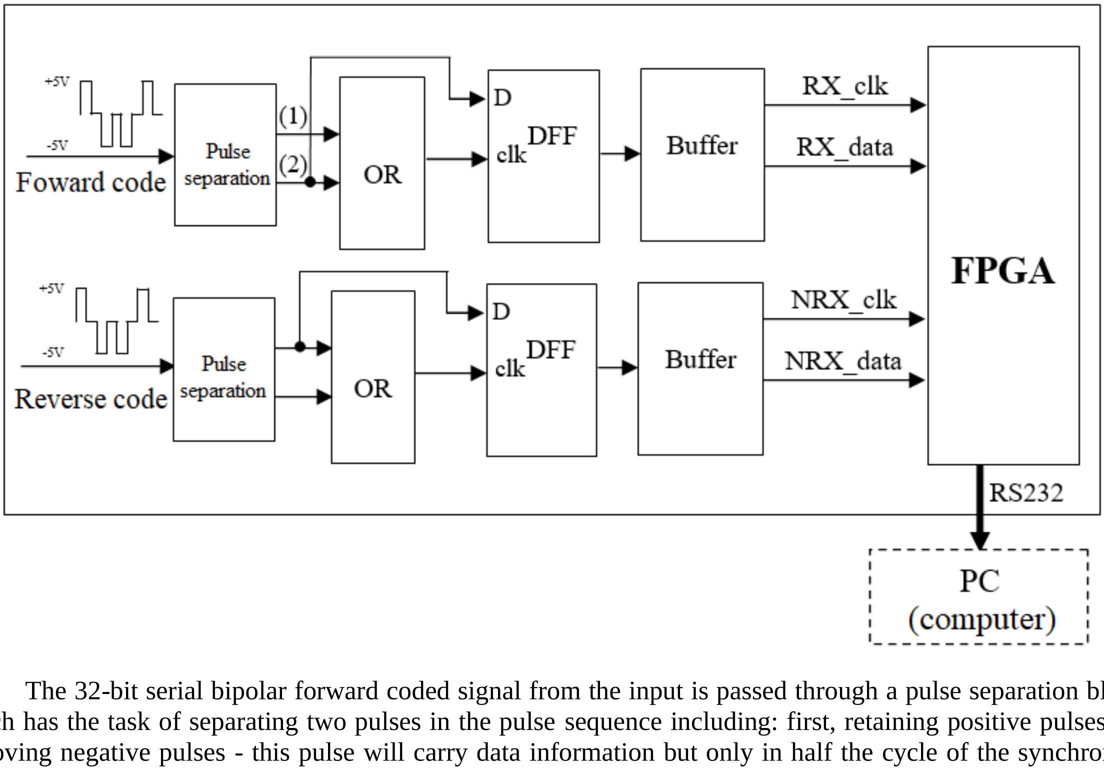
2.1. Processing input signals

The two forward and reverse coded signals delivered to the missile from the new generation aircraft are bipolar signals. Therefore, to be compatible with subsequent processing lines, these two signals must be preliminarily processed into unipolar signals. The subsequent line can use various technologies (digital ICs, microprocessors, FPGA, etc.) to work directly with these unipolar signals. Here, the algorithm being developed is aimed at application to hardware using FPGA (Field Programmable Gate Array) technology.

In the functional diagram (Figure 2), the hardware is divided into two parts. The first part converts bipolar forward and reverse coded signals into synchronous pulses and unipolar data pulses. This part is designed based on pulse separation blocks and flip-flops, buffer circuits. The second part is the data processing section using FPGA technology, which has the task of decoding the data sequence from two pairs of input synchronous pulse and data pulse signals, then storing them as binary files or packaging and transmitting them to a computer. The implementation of the data decoding algorithm is performed using VHDL Language (Very High-Speed Intergrated Circuit Hardware Description Language), then compiled, packaged, and installed in a FPGA chip.

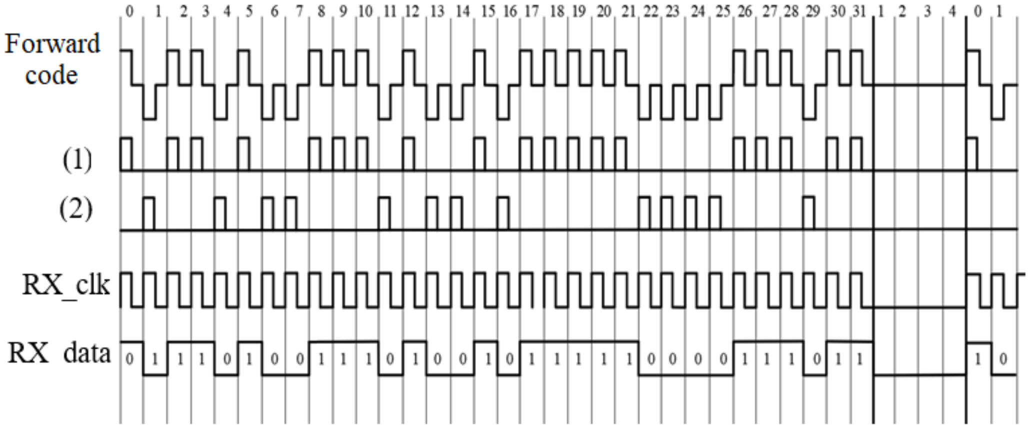
The new generation aircraft communicates with the missile through both forward and reverse transmission paths with the same information to increase reliability in transmitting and receiving command codes. The conversion of forward and reverse codes is completely identical, so we only analyze the conversion of the forward code.

Figure 2. Hardware functional diagram for processing input pulse signals



The 32-bit serial bipolar forward coded signal from the input is passed through a pulse separation block, which has the task of separating two pulses in the pulse sequence including: first, retaining positive pulses and removing negative pulses - this pulse will carry data information but only in half the cycle of the synchronous pulse (*data*); second, removing positive pulses and retaining negative pulses then inverting them. These two pulses continue to be passed through an "OR" gate to create a clock pulse (*clk*). The two clock pulse (*clk*) and data pulse (*data*) signals are used as synchronous pulses and data pulses at the D-type flip-flop stage, with the output being data for the entire cycle. Next, the synchronous and data pulses are passed through a level buffer stage to supply the FPGA inputs. Thus, the separation of bipolar pulses carrying 32-bit information into 50KHz frequency synchronous pulses and 32-bit data sequences is completed. The pulse separation diagram is presented in Figure 3.

Figure 3. Diagram of synchronous pulse and data separation from 32-bit bipolar code



2.2. FPGA Input Pulse Diagram

Thus, the two forward and reverse coded signals delivered from the WCS system of the new generation aircraft are preprocessed by pulse separation blocks. The output of each pulse separation block after passing through buffers consists of two unipolar pulse signals. The pulse separation blocks are completely designed using passive components including resistors, capacitors, digital ICs, and operational amplifier ICs. Since processing forward and reverse coded signals is completely similar, we only consider the forward coded signal. Recognition of both forward and reverse coded signals will allow improvement of data decoding accuracy and reduction of noise-related risks. Thus, the key to the problem is building an algorithm to decode data from two input pulse signals (which are the two output signals from the pulse separation block): synchronous pulses (called RX_clk) and pulses carrying 32-bit data information (called RX_data pulses), (see figure 3).

The RX_clk signal has the following characteristics: the signal consists of bursts of 32 pulses (corresponding to each word) with cycle T=20μs, duty cycle δ=50%, transmitted continuously with rest time between two consecutive pulse series (which is the spacing time between two code words, T pause) of 4T. The RX_data signal carries information about 32-bit data corresponding to 32 pulses encoding 32 bits in the bipolar coded signal (forward code).

2.3. Algorithm for Decoding 32-bit Data from Input Pulse Set

The algorithm for decoding 32-bit data from the input pulse set is built on the basis of a State Machine. Data decoding is performed by searching for the PAUSE pulse (negative pulse with width 4T). An input clk pulse counter (50MHz) and a state machine are established. To confirm that the PAUSE pulse appearance is certain, the counter will count to the count corresponding to half the PAUSE pulse time (count=8**CLKS_PER_BIT*), point (0) on the algorithm flowchart (Figure 4.b). The constant *CLKS_PER_BIT* is the number of standard clock pulses in one cycle (T) of the RX_clk synchronous pulse. If during the PAUSE pulse search process, level 1 of the RX_clk signal appears (point 4 on the algorithm flowchart), then the PAUSE pulse search ends, the state machine resets the counter and returns to the initial state to search for another PAUSE pulse.

When the pause pulse has certainly appeared, the state machine continues to search for the rising edge of the input RX_clk synchronous pulse. To confirm with certainty the high level of the input synchronous pulse, the state machine must probe to the middle point of level 1 of the RX_clk pulse (point 3 on the algorithm flowchart). At this time, similar to searching for the PAUSE pulse, the counter needs to count to the count CLKS_PER_BIT.

Data sampling is performed when the state detector reaches the middle point of level 1 of the RX_clk pulse (point 3, Figure 4.b). This ensures data reliability and high noise immunity.

Next, the state machine continues to probe the low level of the RX_clk pulse based on the falling edge of this pulse, similarly to probing the high level.

Thus, after 1 pulse from the end time of the PAUSE pulse, the first data bit (bit 0) of the 32-bit data sequence is decoded. This process repeats until the data bit index counter reaches 31. At this time, one 32-bit data code word has been completely decoded (DONE state). The state machine resets the counters and returns to the initial state (IDLE) to search for the next PAUSE pulse.

Figure 4.a Algorithm for decoding 32-bit data

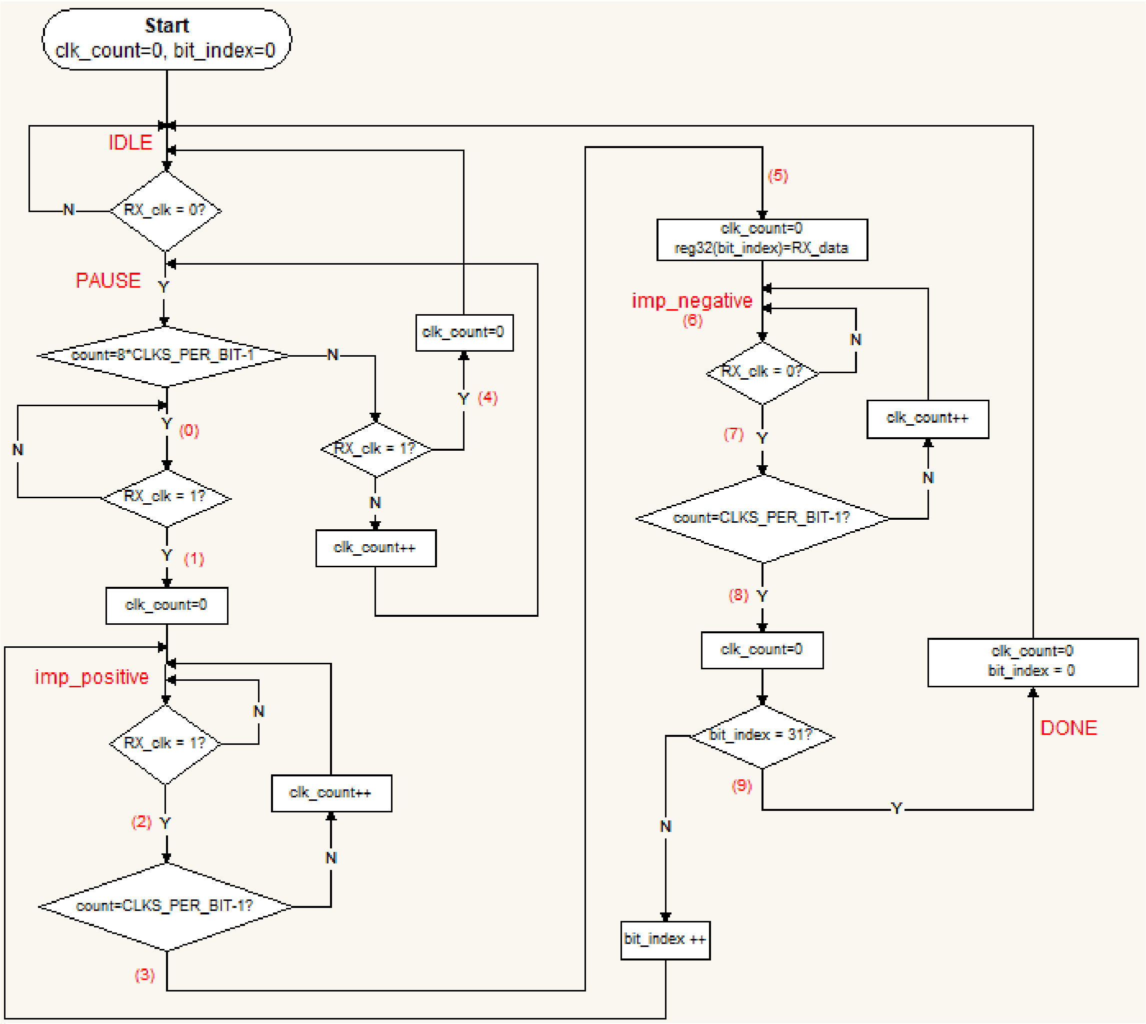
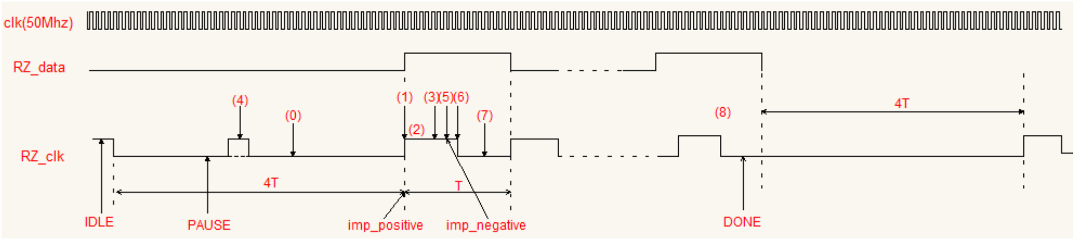


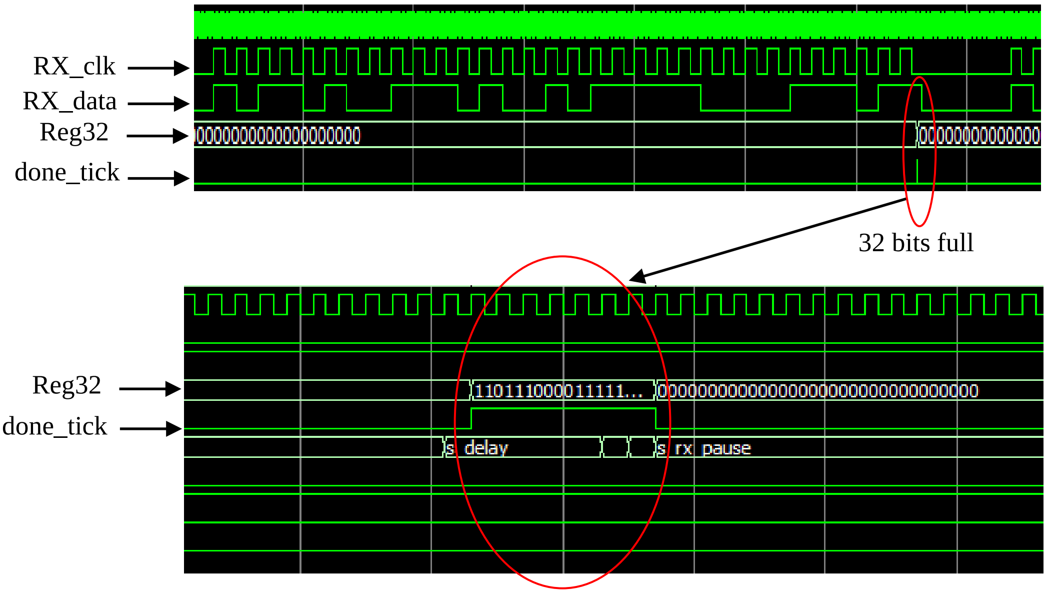
Figure 4.b State diagram describing the algorithm



III. SIMULATION AND EXPERIMENTAL RESULTS

To verify the algorithm, ModelSim software from MentorGraphics was used. The simulation process creates fake 32-bit code words transmitted continuously (represented by the two input *RX_clk* and *RX_data* pulses in Figure 3) according to the correct timing diagram of forward and reverse bipolar coded signals. The results obtained after decoding are 32-bit code words that match the transmitted code words, demonstrating algorithm accuracy.

Figure 5. Image showing simulation results on ModelSim software



Experimental Results:

With the theoretical results, the author has designed and manufactured the “Command decoding block” located in the AA-10 training version missile.

Verification testing was conducted with Su-30 aircraft. The “Command decoding block” located in the AA-10 training version missile is mounted on the aircraft through the “АПУ-470М” launcher (figure 7.a). During operation, the weapon control system on the aircraft will transmit command code information and target indication parameters down to the missile through the serial bipolar coded transmission line into the “Command decoding block”.

Subsequently, the decoded data will be transmitted down to the ground decoding device for online display on the software interface (figure 7.b).

Figure 6. Image of electronic panel of the command decoding block

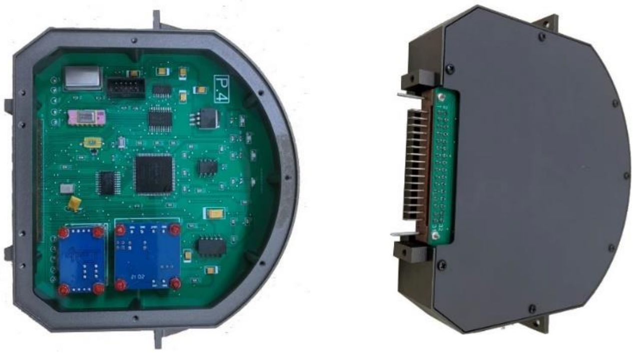
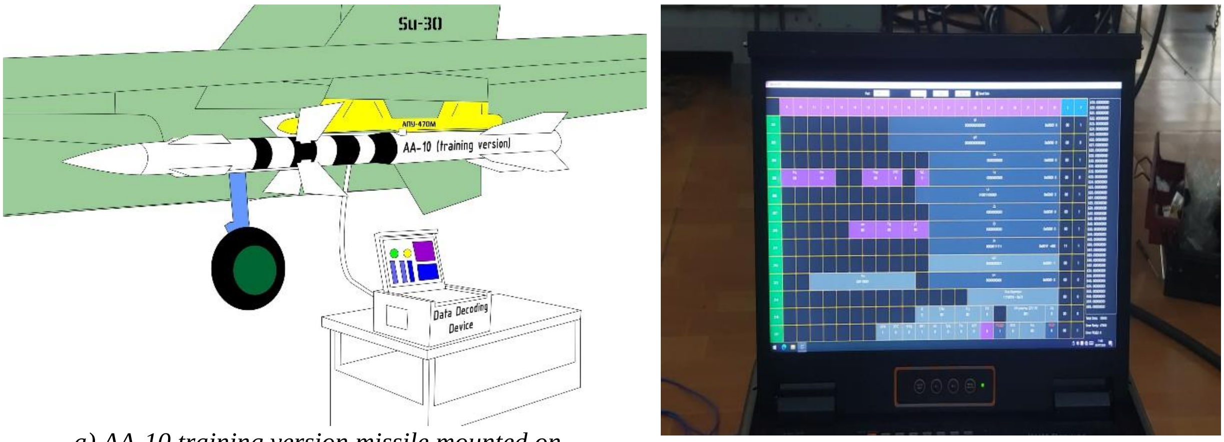


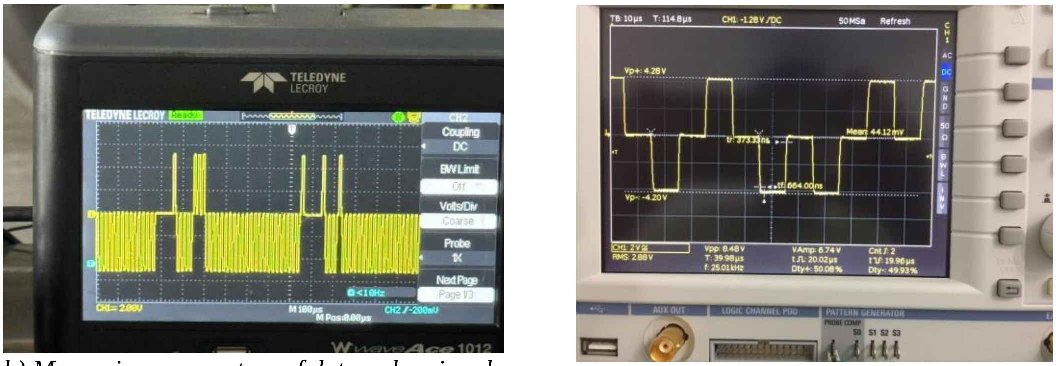
Figure 7. Images of AA-10 training version missile and ground decoding device



a) AA-10 training version missile mounted on Su-30 aircraft for testing

b) Using computer to display decoding results

Figure 8. Images of the process of measuring the bipolar code signal transmitted from Su-30 aircraft to AA-10 missile (training version)

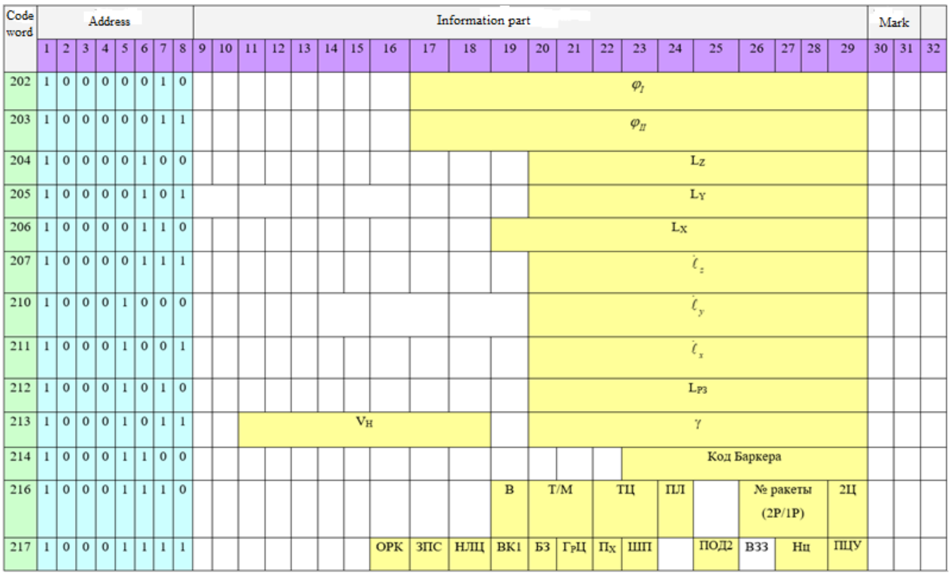


b) Measuring parameters of data pulse signals from aircraft

c) Determining the cycle of data pulse signals from aircraft

The testing process successfully decoded the data structure from Su-30 aircraft transmitted to AA-10 missile as follows: data is structured into 13 code words with addresses: “202”, “203”, “204”, “205”, “206”, “207”, “210”, “211”, “212”, “213”, “214”, “216”, “217”. The transmission of code words is repeated in cycles of 4T=80µs. The structure of each code word consists of 32 bits containing information about simulated target indication data from the new generation aircraft. The structure of bits in the code words has been presented in Figure 9. These results completely match the theoretical research.

Figure 9. Results of decoding data transmitted from new generation aircraft to AA-10 missile



Code word	Address								Information part																					Mark		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
202	1	0	0	0	0	0	1	0									φ_I															
203	1	0	0	0	0	0	1	1									φ_{II}															
204	1	0	0	0	0	1	0	0												L_Z												
205	1	0	0	0	0	1	0	1												L_Y												
206	1	0	0	0	0	1	1	0											L_X													
207	1	0	0	0	0	1	1	1												$\dot{\ell}_z$												
210	1	0	0	0	1	0	0	0												$\dot{\ell}_y$												
211	1	0	0	0	1	0	0	1												$\dot{\ell}_x$												
212	1	0	0	0	1	0	1	0												$L_{РЗ}$												
213	1	0	0	0	1	0	1	1			V_H									γ												
214	1	0	0	0	1	1	0	0															Код Баркера									
216	1	0	0	0	1	1	1	0											В	Т/М		ТЦ		ПЛ		№ ракеты (2Р/1Р)			2Ц			
217	1	0	0	0	1	1	1	1								ОРК	ЗПС	НЛЦ	ВК1	БЗ	ГрЦ	Пх	ШП		ПОД2	ВЗЗ	Нц		ПЦУ			

IV. CONCLUSION

The paper has presented a research method for developing algorithms to decode data in transmission lines from carrier aircraft to the several types of radio-guided missiles. The author started from studying the characteristics of bipolar coded signals transmitted down from carrier aircraft to radio-guided missiles to research and develop transmission line data decoding algorithms. Verification results on ModelSim simulation software from MentorGraphics show the correctness of the algorithm. The test was carried out with Su-30 aircraft and AA-10 missile. The algorithm has been used to design and manufacture command decoding blocks, self-recording blocks, etc., serving the process of designing and manufacturing training missiles for combat application training.

REFERENCES

[1]. Air Defense-Air Force Command, *"Technical description of the RGS-27 radio homing head"*, Ha Noi, 2015.

[2]. Nguyen Tan Phuoc, *"Basic and Advanced pulse techniques"*, Ho Chi Minh City Publishing House, Ho Chi Minh City, 2002.

[3]. PONG P.CHU, *"FPGA prototyping by verilog examples"*, John Wiley&Son, Inc., Hoboken, New Jersey, 2008.

[4]. Tong Van On, *"Digitl Circuit Design With VHDL&Verilog"*, Social Labor Publishing House, Ha Noi, 2007.

[5]. The MentorGraphics, ModelSim User's Manual, March 2015.