

данное действие, операции объединены в группы. Каждой из них присвоен код, который используется для описания управляющих сигналов. Каждое условие аналогично имеет свой код.

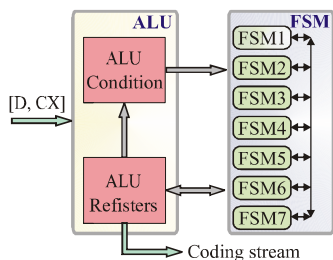


Рис. 3. Структура аппаратной модели MQ-кодера

Таблица 1. Базовые операции регистров MQ-кодера

Initialization operations	
Code	Operation
a0	A = 0x00008000;
c0	C = 0x00000000;
bp0	BP = BPST - 1;
ct0	CT = 12;
ct1	CT = 13;
i0	reset I(CX)
mps0	reset MPS(CX)
Basic operations	
a1	A = Qe(I(CX));
a2	A = A - Qe(I(CX));
a3	A = A << 1;
b1	B = B + 1;
b2	B = C >> 19;
b3	B = C >> 20;
bp1	BP = BP + 1;
c1	C = C & 0x000FFFFF;
c2	C = C & 0x0007FFFF;
c3	C = C & 0x07FFFFFF;
c4	C = C + Qe(I(CX));
c5	C = C - 0x00008000;
c6	C = C 0x0000FFFF;
c7	C = [0x0000, (c(15:0) - 0x8000)];
c8	C = C << 1;
c9	C = C << CT;
mps1	MPS(CX) = 1 - MPS(CX);
i1	I(CX) = NLPS(I(CX));
i2	I(CX) = NMPS(I(CX) ;
ct2	CT = 7;
ct3	CT = 8;
ct4	CT = CT - 1;
dis_b	discard B;

Таблица 2. Условия алгоритма MQ-кодера

Conditions	
Code	Operation
e1	D == MPS(CX)
e2	B == 0xFF
e3	switch(I(CX)) == 1
e4	CT == 0
e5	B == 0xFF
n1	A < Qe(I(CX))
n2	A >= Qe(I(CX))
n3	A < 0x8000
n4	C < 0x08000000
n5	C(15:0) >= C+A

3. Архитектура АЛУ

АЛУ регистров содержит компоненты: A, C, I, MPS, CT, B, BP. Кроме того, устройство содержит счетчик L, подсчитывающий число байтов кода, сгенерированного на данный момент; буфер T временных данных. Вычитающий счетчик CT идентифицирует точку (=0), в которой частично генерируемые биты кода переносятся из регистра C в буфер временных байтов T. В АЛУ также содержатся таблицы константных значений: Qe, NMPS, NLPS.

Модели операционных блоков регистров A, C, BP, B, CT могут быть представлены с помощью автомата Мура, класса C [16] (рис. 5), который определяется вектором $M = (S, Z, W, \varphi, \psi, a_1)$, где $S = (s_1, \dots, s_m, \dots, s_M)$ – множество состояний и $W = (w_1, \dots, w_g, \dots, w_G)$ – совокупность выходных сигналов, идентичных S, соответствующих множеству значений, принимаемых устройством; $Z = (z_1, \dots, z_m, \dots, z_M)$ – совокупность входных сигналов: значения данных и управляющие сигналы, определяющие выбор операции; $s(t+1) = \varphi(s(t), z(t))$ – функция переходов; $\psi: w(t) = s(t)$ – функция выходов.

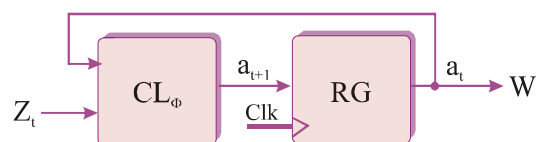


Рис. 5. Автомат Мура класса C

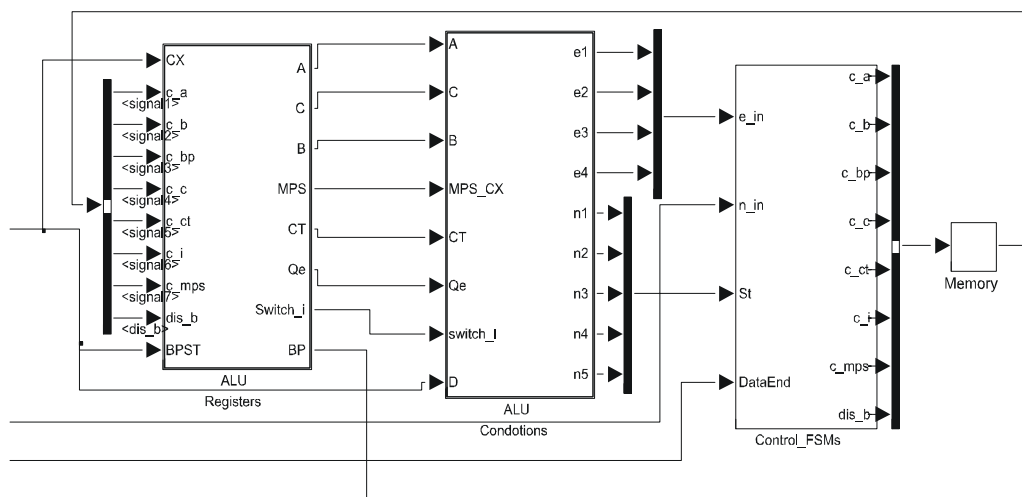


Рис. 4. Simulink-модель MQ-кодера

Модель операционного автомата для регистра А (рис. 6) представлена функциями переходов $s(t+1) = \varphi(s(t), z(t))$ и выходов $w(t) = s(t)$, где $S = W = (0, \dots, 2^{32} - 1)$ – множество состояний и значений выходов; $Z = (a_1, a_2, a_3, a_4, Qe(I(CX)))$ – множество входных сигналов.

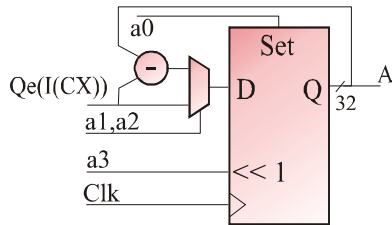


Рис. 6. Регистр А

Модель операционного автомата для регистра С (рис. 7) представлена функциями переходов $s(t+1) = \varphi(s(t), z(t))$ и выходов $w(t) = s(t)$, где $S = W = (0, \dots, 2^{32} - 1)$ – множество состояний и значений выходов;

$Z = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, Qe(I(CX)), CT)$ – множество входных сигналов.

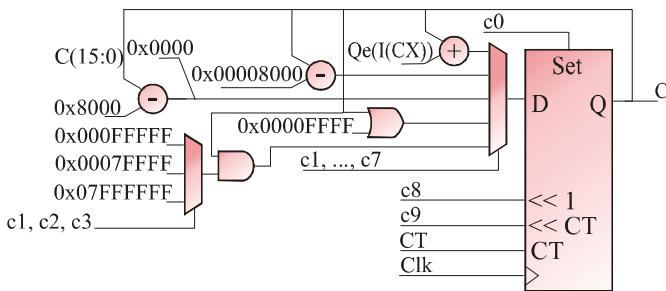


Рис. 7. Регистр С

Модель операционного автомата для регистра СТ (рис. 8) представлена функциями переходов $s(t+1) = \varphi(s(t), z(t))$ и выходов $w(t) = s(t)$, где $S = W = (0, \dots, 13)$ – множество состояний и значений выходов; $Z = (ct_0, ct_1, ct_2, ct_3)$ – множество входных сигналов.

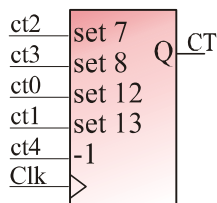


Рис. 8. Регистр СТ

Модель операционного автомата для регистра В (рис. 9) представлена функциями переходов $s(t+1) = \varphi(s(t), z(t))$ и выходов $w(t) = s(t)$, где $S = W = (0, \dots, 2^{16} - 1)$ – множество состояний и значений

выходов; $Z = (b_1, b_2, b_3, C)$ – множество входных сигналов.

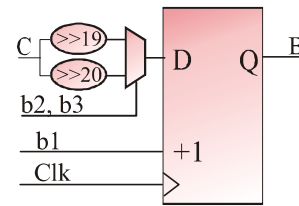


Рис. 9. Регистр В

Модели операционных автоматов для регистров MPS и I представляют собой классический автомат Мили, поскольку состояние устройства реализуется с помощью памяти с асинхронным чтением. Модель регистра MPS (рис. 10) представлена функциями переходов $s(t+1) = \varphi(s(t), z(t))$ и выходов $w(t) = \psi(s(t), CX)$, где $S = (0, \dots, 2^{19} - 1)$ – множество состояний; $W = (0, 1)$ – множество значений выходов; $Z = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, Qe(I(CX)), CT)$ – входные сигналы.

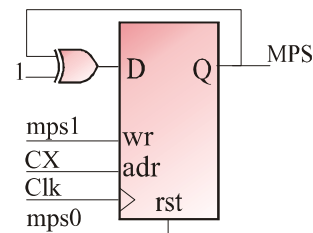


Рис. 10. Регистр MPS

Модель операционного автомата для регистра I (рис. 11) представлена функцией переходов $s(t+1) = \varphi(s(t), z(t))$ и выходов $w(t) = \psi(s(t), CX)$, где $S = (a_0, \dots, a_{19})$, $a \in (0, \dots, 46)$ – множество состояний; $W = (0, 46)$ – множество значений выходов; множество входных сигналов $Z = (i_0, i_1, i_2, CX, NLPS(I), NMPS(I))$.

В программе Simulink модели регистров А (рис. 12, а) С, В, ВР реализуются М-функциями с элементами задержки z^{-1} , а модели регистров I (рис. 12, б) и MPS – блоками памяти, описанными S-функциями, для регистров.

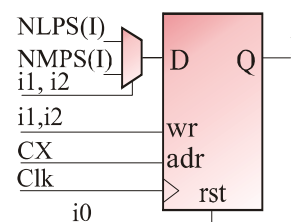


Рис. 11. Регистр I

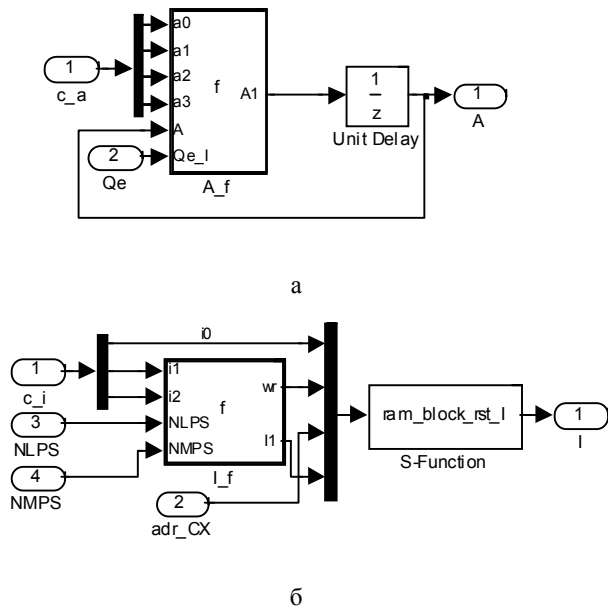


Рис. 12. Simulink-модели регистров

На рис 13. представлена Simulink-модель АЛУ регистров, соответствующая блоку ALU registers с рис. 4.

АЛУ для вычисления условий реализуется с помощью комбинационной схемы, где входными переменными являются выходы регистров из соответствующего АЛУ (рис. 14).

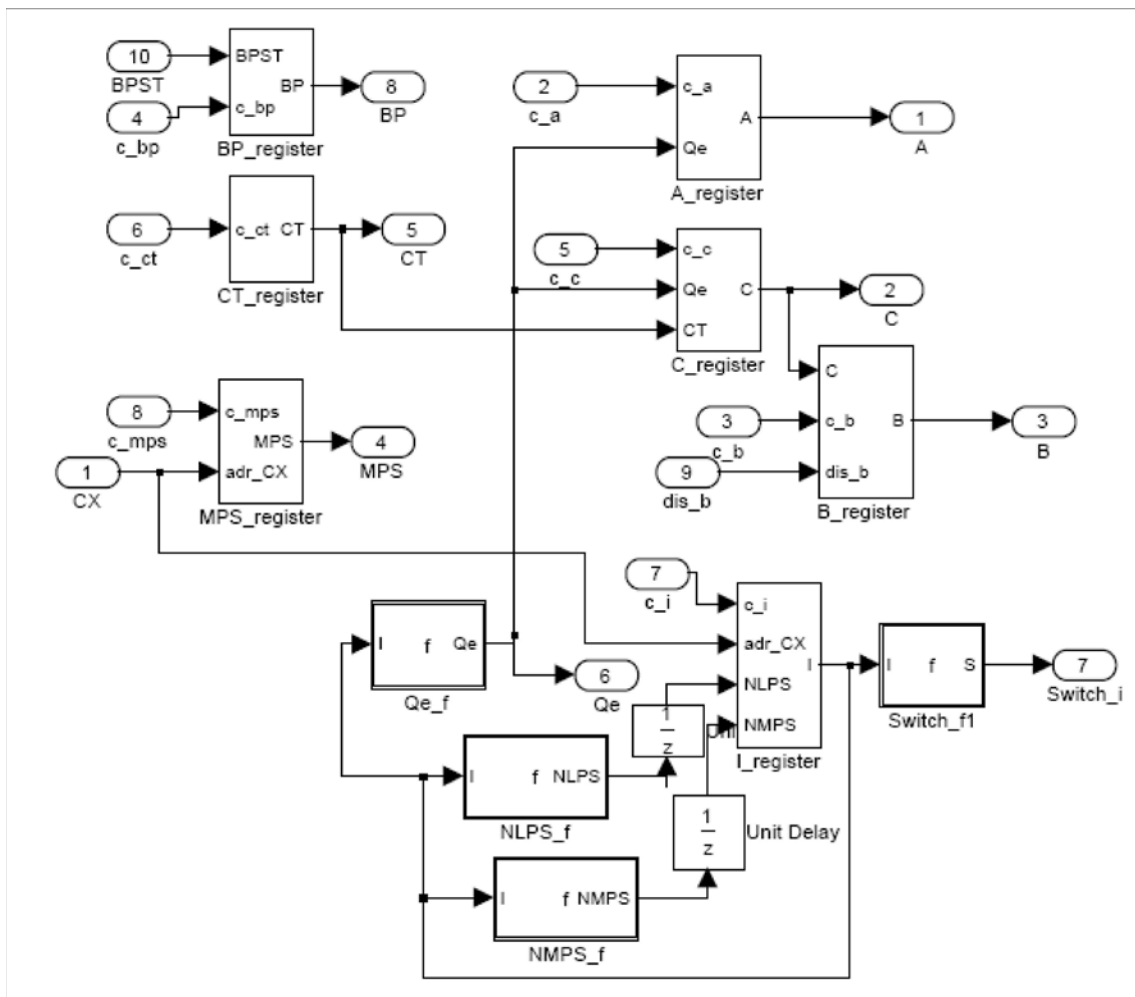


Рис. 13. Модель АЛУ регистров

4. Сеть автоматов управляющего блока

Управляющий блок реализуется с помощью иерархических автоматов, построенных на основе взаимосвязанных процедур алгоритма MQ-кодера (см. рис. 2). Модель такого устройства (рис. 15) представляет собой асинхронный автомат, в котором каждый компонент функционирует независимо от остальных и взаимодействует с последними только с помощью управляющих сигналов, инициирующих начало и окончание процедуры. Иерархический управляющий блок является сетью автоматов [17], модель которой задается вектором: $N = (Z, \{S_i\}, W, \{f_i\}, \{\psi_i\}, g)$, где Z – входной алфавит, $\{S_i = (A_i, Z_i, \delta_i)\}$, $1 \leq i \leq n$ – множество компонентных автоматов (КА) (рис 16) сети:

$$Z_i = \begin{cases} Z_i' \times Z_i'' & \leftarrow Z_i' \neq \emptyset; \\ Z_i'' & \leftarrow Z_i' = \emptyset, \end{cases}$$

где Z_i' , Z_i'' – внутренний и внешний входные алфавиты S_i . Функция переходов $S_i \delta_i : A_i \times Z_i \rightarrow A_i$; W – выходной алфавит сети; $\{f_i : (\times A_j) \rightarrow Z_i'\}$, $1 \leq i, j \leq n$ – множество функций соединения КА сети; $\{\psi_i : Z \rightarrow Z_i'\}$, $1 \leq i \leq n$ – множество входных функций; $g : (\times A_i) \times Z \rightarrow W$ – выходная функция сети. Двух-компонентное множество $\{\{S_i\}, \{f_i\}\}$ является базисом и структурой проектируемой сети.

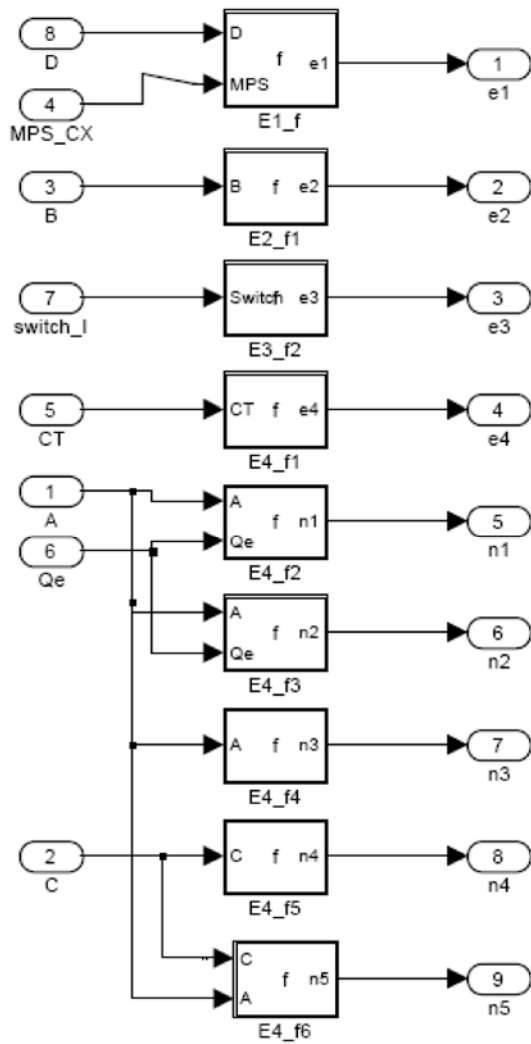


Рис. 14. Модель АЛУ условий

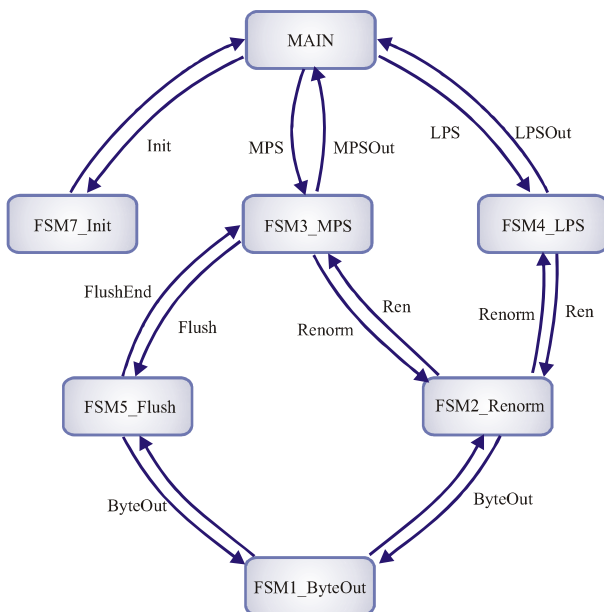


Рис. 15. Модель асинхронного иерархического управляющего автомата

Алгоритм функционирования асинхронного иерархического управляющего автомата задается таблицей

переходов (табл. 3). На параллельность функционирования такого устройства указывает наличие переходов в различные состояния при одном и том же условии. Например, компонентный автомат FSM1_ByteOut передает управление сразу двум вершинам FSM5_Flush и FSM2_Renorm.

Таблица 3. Таблица переходов иерархического автомата

Текущее состояние	Условие	Следующее состояние
MAIN	Init	FSM7_Init
	MPS	FSM3_MPS
	LPS	FSM4_LPS
FSM1_ByteOut	-	FSM5_Flush
	-	FSM2_Renorm
FSM2_Renorm	Ren	FSM3_MPS
	Ren	FSM2_LPS
FSM3_MPS	MPSOut	MAIN
FSM4_LPS	LPSOut	MAIN
FSM5_Flush	FlushEnd	FSM3_MPS
FSM7_Init	-	MAIN

В табл. 3 и на рис.15 отсутствие условия при выполнении перехода означает безусловный переход. В этом случае оба автомата – ведущий и ведомый функционируют параллельно, до тех пор пока второй из них не выполнит свой алгоритм и не перейдет в начальное состояние, ожидая очередного сигнала вызова. Эта ситуация возможна, если ведомый автомат имеет небольшое количество состояний и выполняет функции, не входящие в конфликт с выполняемыми в этот же период времени функциями ведущего автомата. Продолжительность выполнения одного цикла алгоритма соответствует наихудшему варианту – самому длинному пути обхода графа. Поскольку реализация автоматов с числом состояний, превышающим 6 - 7, является неэффективной из-за формирования длинных путей комбинационных функций и, как следствие, снижения частоты функционирования устройства, то использование асинхронной модели позволяет реализовывать сложные автоматы, сохраняя высокую допустимую частоту синхросигнала.

Таблицами 4-10 представлены множества переходов КА, составляющих сеть автоматов, формирующих иерархический асинхронный управляющий блок.

Таблица 4. Таблица переходов компонентного автомата FSM1_ByteOut

FSM1_ByteOut			
a_i	Z_i	a_{i+1}	w_i
Start	ByteOut&e2'&n4'	S1	b1
	ByteOut&e2	Start	b3, c1, ct2, bp1
	ByteOut&e2'&n4	S1	b2, c4, ct3, bp1
	ByteOut'	Start	-
S1	e2	Start	b3, c1, ct3, bp1
	e2'	Start	b2, c4, ct3, bp1

Для разработки исполняемой модели управляющего блока использовался инструмент Stateflow программного пакета Simulink среды Matlab (рис. 16). Этот редактор предоставляет возможность для создания параллельных иерархических состояний и моделирования их поведения в рамках единого устройства.

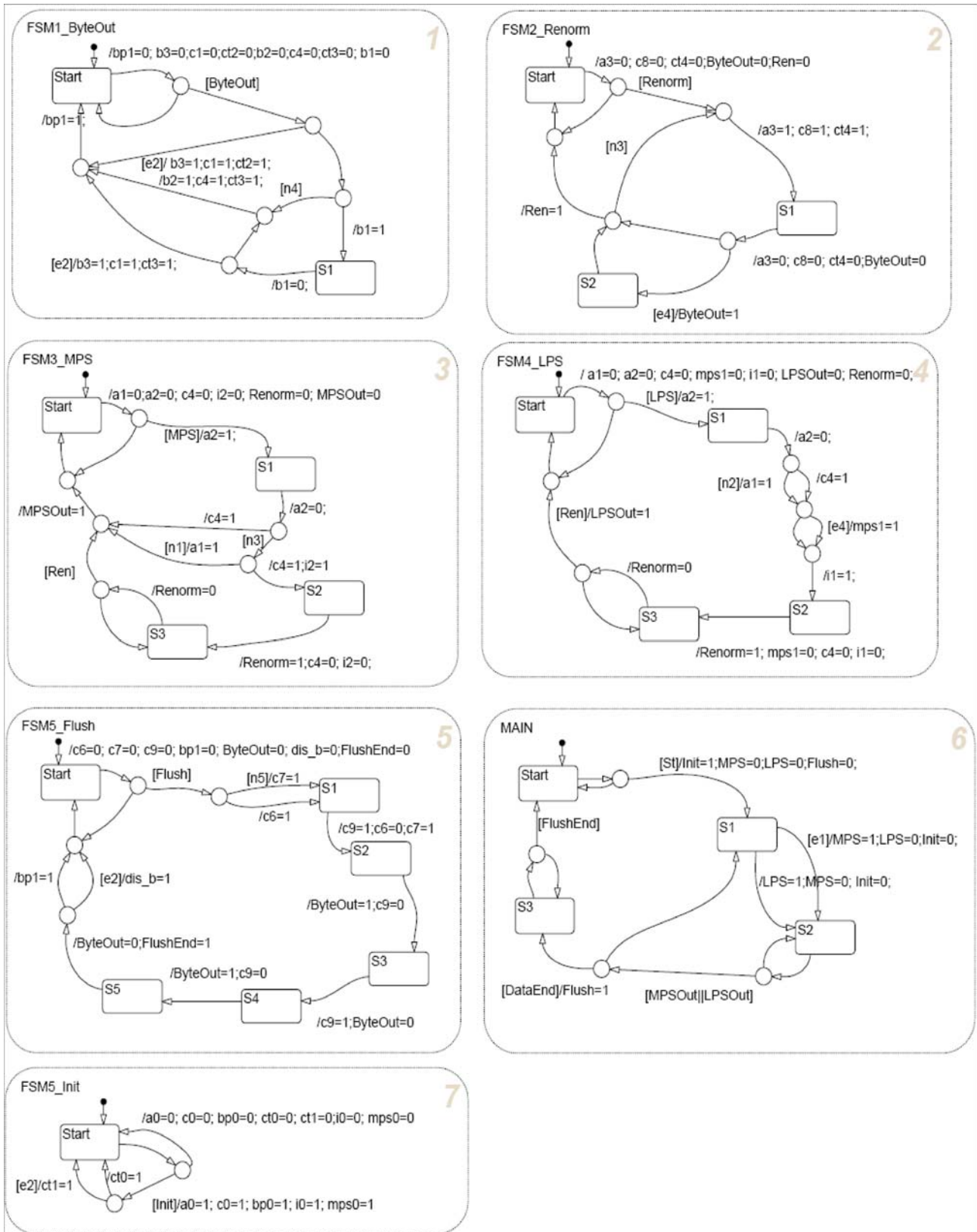


Рис. 16. Модель асинхронного управляющего автомата

Таблица 5. Таблица переходов компонентного автомата FSM5_Init

FSM5_Init			
a_i	Z_i	a_{i+1}	w_i
Start	Init & e2	Start	a0, c0, bp0, i0, mps0, ct1
	Init & e2'	Start	a0, c0, bp0, i0, mps0, ct0

Таблица 6. Таблица переходов компонентного автомата FSM2_Renorm

FSM2_Renorm			
a_i	Z_i	a_{i+1}	w_i
Start	Renorm	S1	a3,c8,ct4
	Renorm'	Start	-
S1	e4	S2	ByteOut
	e4'&n3	S1	a3,c8,ct4
	e4'&n3'	Start	Ren
S2	n3	S1	a3,c8,ct4
	n3'	Start	Ren

Таблица 7. Таблица переходов компонентного автомата FSM3_MPS

FSM3_MPS			
a_i	Z_i	a_{i+1}	w_i
Start	MPS	S1	a2
	MPS'	Start	-
S1	n3&n1'	S2	a2,c4,i2
	n3'	Start	a2,c4, MPSOut
	n3&n1	Start	a1,a2, MPSOut
S2	-	S3	c4, i2, Renorm
S3	Ren	Start	MPSOut
	Ren'	S3	-

Таблица 8. Таблица переходов компонентного автомата FSM4_LPS

FSM4_LPS			
a_i	Z_i	a_{i+1}	w_i
Start	LPS	S1	a2
	LPS'	Start	-
S1	n2&e4	S2	a1, a2, mps1, i1
	n2'&e4		a2, c4, mps1, i1
	n2&e4'		a1, a2, i1
	n2'&e4'		a2, c4, i1
S2	-	S3	Renorm
S3	Ren	Start	LPSOut
	Ren'	S3	

Таблица 9. Таблица переходов компонентного автомата FSM5_Flash

FSM5_Flash			
a_i	Z_i	a_{i+1}	w_i
Start	Flush&n5	S1	c7
	Flush&n5'	S1	c6
	Flush'	Start	
S1	-	S2	c6, c7, c9
S2	-	S3	ByteOut
S3	-	S4	c9
S4	-	S5	ByteOut
S5	e2	Start	FlushEnd, dis_b
	e2'	Start	bp1

Таблица 10. Таблица переходов компонентного автомата MAIN

MAIN			
a_i	Z_i	a_{i+1}	w_i
Start	St	S1	Init
	St'	Start	-
S1	e1	S2	MPS
	e1'	S2	LPS
S2	(LPSOut&MPSOut)DataEnd	S3	Flush
	(LPSOut&MPSOut)DataEnd'	S1	-
	LPSOut' & MPSOut'	S2	-
S3	FlushEnd	Start	-
	FlushEnd'	S3	-

5. Заключение

Научная новизна: разработана новая архитектура аппаратной реализации MQ-кодера, основанная на асинх-

ронном иерархическом управляющем автомате, состоящем из 7 параллельно работающих компонентов, которая дает возможность упростить реализацию функций переходов и выходов в целях повышения быстродействия MQ-кодера (в 4-6 раз) по сравнению с аналогичными автоматами, предложенными Andra, Chakrabarti и Acharya в [15, 18-21].

Практическая значимость: предложенная архитектура MQ-кодера является аппаратным компонентом для блока энтропийного кодирования устройства сжатия на основе стандарта JPEG2000, использующего асинхронный способ соединения компонентов.

Литература: 1. www.mathworks.com. 2. Дьяконов В.П. MATLAB 6.5 SP1/7 + Simulink 5/6. Основы применения. Серия «Библиотека профессионала». М.: СОЛОН Пресс.– 2005. 800 с. 3. Дьяконов В. П. MATLAB 6.5 SP1/7.0 + Simulink 5/6. Обработка сигналов и проектирование фильтров. М.: Наука, 1999. 561 с. 4. Дьяконов В. П., Кружлов В. Математические пакеты расширения MATLAB. Специальный справочник. СПб.: Питер, 2001. 480 с. 5. Simulink. Simulation and Model-Based Design. Writing S-Function. Version 7. 2005. 322 p. (www.mathworks.com) 6. Simulink. Simulation and Model-Based Design. Getting Started. Version 6. 2005. 37 p. (www.mathworks.com) 7. Stateflow. For Use with Simulink. Getting Started. Version 7. 2005. 184 p. (www.mathworks.com) 8. David S. Taubman, Michael W. Marcellin. JPEG2000: image compression fundamentals, standards and practice. Boston/Dordrecht/London: Kluwer Academic Publishers. 2002. 774 p. 9. Миано Дж. Форматы и алгоритмы сжатия изображений в действии. Учебн. пособ. М.: Издательство Триумф. 2003. 336 с. 10. Сэлмон Д. Сжатие данных, изображений и звука. М.: Техносфера. 2004. 368 с. 11. ISO/IEC 15444-1.– Information Technology-JPEG2000 Image Coding System-Part 1: Core Coding System. 2000. 211 p. 12. ISO/IEC 15444-2. Final Committee Draft. Information Technology-JPEG2000 Image Coding System. Part 2: Extensions. 2000. 198 p. 13. Tinku Acharya, Ping-Sing Tsai. JPEG2000 Standard for Image Compression. Concepts, Algorithms and VLSI Architectures.– John Wiley & Sons, Inc. 2005. 292 p. 14. Serge Leef. A Methodology for Virtual Hardware/Software Integration. Mentor Graphics Corporation white paper. 2003. 6 p. 15. Taubman D. High Performance Scalable Image Compression with EBCOT. IEEE Trans. IP.– Vol. 9. No. 7. 2000. P. 1158-1170. 16. Соловьев В.В. Проектирование цифровых автоматов на основе программируемых логических интегральных схем. М. Горячая линия. Телеком. 2001. 636 с. 17. Баранов С.И. Синтез микропрограммных автоматов. Л: Энергия. 1079. 232 с. 18. Andra K., Acharya T., and Chakrabarti C. A High Performance JPEG2000 Architecture // Proc. of the IEEE Intl. Symposium on Circuits and Systems (ISCAS 2002). Scottsdale, Arizona. May 2002. P. 765-768. 19. Andra K., Chakrabarti C., Acharya T. A High Performance JPEG2000 Architecture. IEEE Transactions of Circuits and Systems for Video Technology. Vol. 13. N. 3. March 2003. P. 209-218. 20. Andra K., Acharya T., Chakrabarti C. Efficient VLSI Implementation of Bit-plane Coder of JPEG2000. Proc. of the SPIE Intl. Symposium on Optical Science and Technology, Applications of Digital Image Processing XXIV. V. 4472. San Diego, July 2001. P. 246-257. 21. Andra K. Wavelet and Entropy Coding Accelerators for JPEG2000. Ph.D. Dissertation. Arizona State University. December 2001. 180 p.

Поступила в редколлегию 29.11.2007

Рецензент: д-р. техн. наук, проф. Кривуля Г.Ф.

Хаханова Ирина Витальевна, докторантка кафедры АПИВТ ХНУРЭ. Научные интересы: проектирование цифровых систем на кристаллах. Увлечения: английский язык, музыка. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanova@mail.ru