A Closed Caption TV Microcontroller

Ekachai Leelarasmee

Chulalongkorn University, EE Dept.
Bangkok 10330, Thailand
Email : ekachai.l@chula.ac.th

Kanitpong Pengwon

Chiang Mai University, EE Dept.
Chian Mai 50200, Thailand
Email : kanitp@eng.cmu.ac.th

Abstract – A simple RISC microcontroller architecture is implemented using FPGA and full custom chips design. This MCU is fast enough to perform all real time TV functions including closed caption decoding and on screen displaying.

I. Introduction

We have designed a simple 16 bit microcontroller (MCU), named “Q”, that can perform all functions found in many commercialized but complicated closed caption (CC) TV MCUs [3-5]. As shown in Fig. 1, our design employs the Harvard architecture using 24 bits program word and simple memory-mapped input/output (I/O) units such as registers and pulse width modulators (PWMs). Its CPU, shown in Fig. 2, is of RISC type having 15 instructions as shown in Table 1. This CPU is implemented in a two stage pipeline capable of executing and fetching in the same clock with its operational state diagram given in Fig.3. All arithmetic/logic instructions are executed in 1 clock cycle whereas all branching instructions take 2 clock cycles. One distinctive feature of this MCU is that the CPU always responds to an interrupt request INT in exactly 2 clocks by going to state #3 no matter if the current state is #1 or #2. This feature is desirable for handling real time asynchronous TV functions.

II. On Screen Display (OSD) Functions

The OSD unit in Fig.1 can display characters (text/ symbol) in 7 possible colors with a size of 16 x 8 dots on the screen as shown in Fig. 4. A data memory of 256x16x8 bits = 4Kbytes, called Character Fonts, stores all possible 8 bit characters. During the OSD period, the CPU continuously puts 8-bit row dots from the Character Fonts in the shift register which operates at 6MHz. At this shifting frequency, the NTSC line resolution is 63.556 μSec x 6 MHz = 381 dots or approximately 47 characters. The dot output is ANDed with a 3 bit color register to generate R(Red), G(Green), B(Blue) outputs to the cathode ray tube of a TV receiver. The CPU needs an asynchronous external interrupt H_sync for timing the OSD display. Since the interrupt response of the Q CPU is always 2 clocks, the timing jitter between different OSD lines can be at most one CPU clock as opposed to 3-6 for other general MCU [1-2]. This OSD timing jitter corresponds to 1/24MHz = 41.6nS or 0.25 dot, which is quite acceptable.

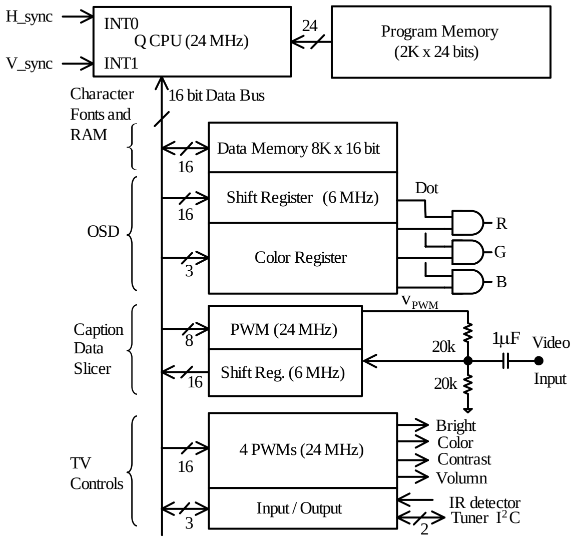

Fig. 1. Top level architecture of a 16 bit Q MCU

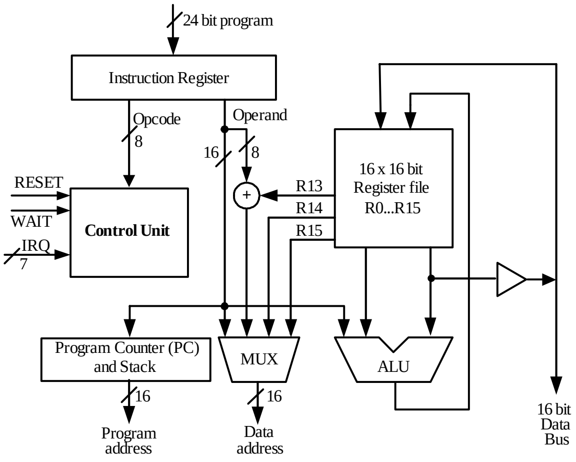

Fig. 2. Internal structure of the Q CPU

Table I Instruction set of the Q MCU

Type	Opcode	#clocks
arithmetic	ADD, ADC,SUB, SBC, ASL	1
logic	AND, OR, XOR, ROL, ROR, SWP	1
branch	JMP, CALL, RET, RETI	2

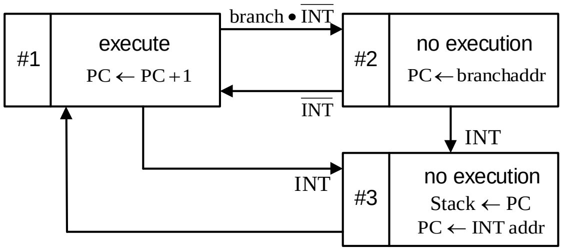

Fig. 3. Operation diagram of the pipelined Q CPU

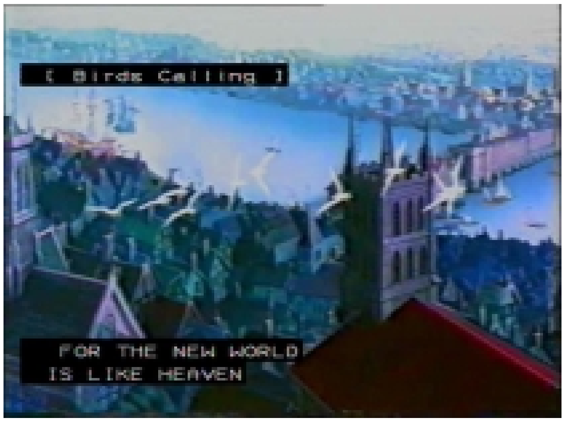

Fig.4. A sample OSD output of the Q MCU

III. USING Q MCU AS AN ADAPTIVE CC DATA SLICER

Data slicing is another critical timing function that can be implemented using simple I/O units connected in Fig.1. Its PWM generates v_{PWM} for shifting the DC level of the video input to make its CC data waveform, shown in Fig.5, swing around the switching level of the shift register which continuously samples at 6Mbit / Sec. However, only sliced data in Line 21 are read for extracting 19 bits of CC data. The CPU then programs the PWM so that its shifting effect would make the duty cycle of the sliced Clock Run-In closer to 0.5. Assuming a 5Volt DC supply, the shifting effect from v_{PWM} is from 0V to 2.5V with an increment of $2.5/(2^8-1) = 10\text{mV}$. The two $20\text{k}\Omega$ resistors and $1\mu\text{F}$ give a time constant of 10mSec, which provides a good AC coupling for the video signal and a good low pass filter for v_{PWM}.

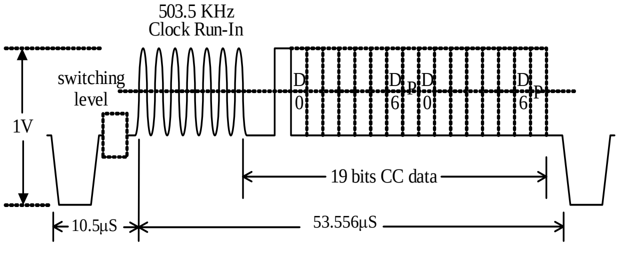

Fig.5. An NTSC Line 21 CC Data Signal Waveform

IV. PROTOTYPES OF THE Q MCU

The Q MCU was implemented with a Xilinx FPGA plus a few off chip memories. It is used to replace a commercial TV MCU in a locally made TV receiver as in Fig.6. Its full custom CMOS design is also laid out and verified with simulation. The whole chip uses 60.8 mm^2 area as follows.

Unit name	**L** (µm)	**Area** (mm^2)	**#Standard cells**
CPU	0.7	8.1	1142
Peripherals	0.7	3.2	805
Data Memory	1.2	12.2	8K x 16 bits
Program Memory	1.2	6.3	2K x 24 bits
Pads	-	10.8	-
Interconnects	-	20.2	-

Fig.6 An FPGA based Q MCU used in a real TV set.

V. CONCLUSION

A high performance RISC MCU can provide all functions of a CC TV receiver without using dedicated hardware such as data capture and OSD found in commercialized TV MCU [3-5]. Synchronization with external events in the video signal is achieved by designing the MCU to have fast and consistent interrupt latency.

ACKNOWLEDGEMENTS

This work is supported by Chulalongkorn University's Ratchadapiseksompote and Thai Government Fund.

REFERENCES

[1] Philips Semi, "80C51-based 8-bit microcontrollers data book", 1995.

[2] Atmel Corporation, "AVR 8-bit RISC microcontrollers", May 1997.

[3] Hynix, "HMS8IC4x60 TV microcontroller", May 2002.

[4] Philips Semiconductors, "SAA55xx TV microcontroller with closed captioning and on-screen display", February 2000.

[5] SGS-Thompson Microelectronics, "ST9294 2K ROM HCMOS MCU wit on screen display and closed caption data slicer", May 1993.