ECES 488 Embedded System

Specification and Design of a Video Phone System

by

Wei Xiang

Liang You

Instructor

Professor C.A. Papachristou

April 2000

System Functions

- If videophone rings and the user responds, the device works normally as a phone.

- If user does not respond, the answering machine is going to be activated and the input data (audio and/or video) will be routed to their respective units.

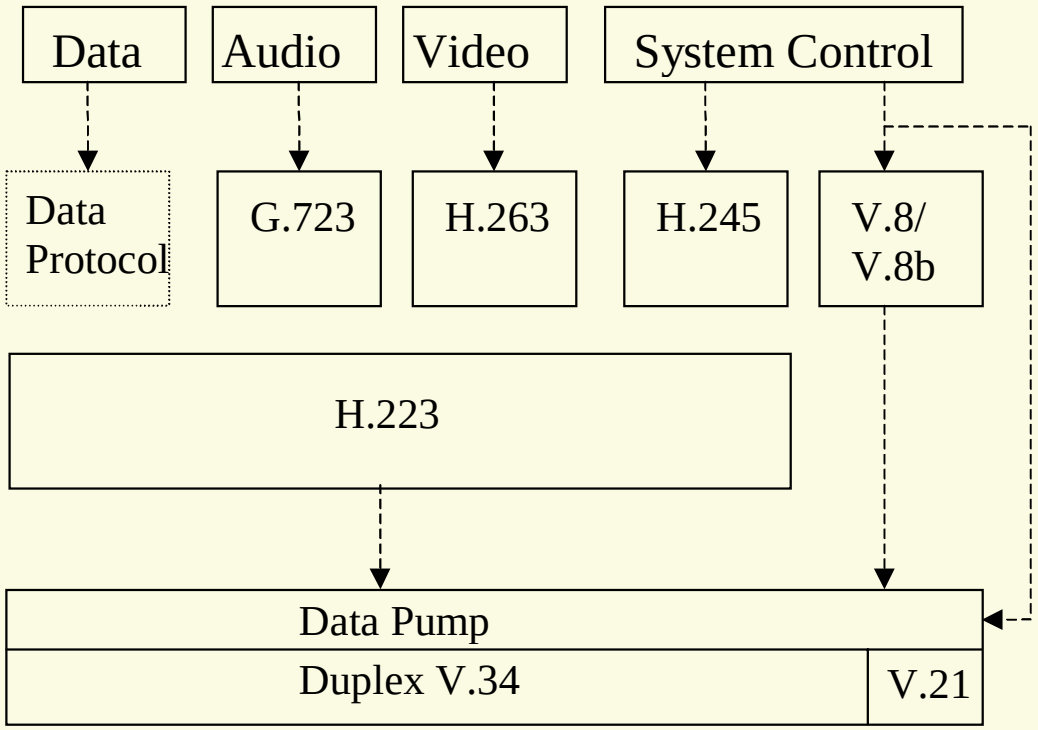
- If during conversation, user decides to send image out, a button can be pressed to ask the CCD camera to capture picture. The image will be processed and sent.

- If the device get an incoming image, it will be channeled to the respective units and the data will be processed, displayed and saved.

System Specification

Power:	supplied by external power supply
Type of Communication:	transmission and receipt of voice +simultaneous video
Transmission Time:	max. 30 fps (frame per second)
Modem:	using built-in modem
Standards:	system ITU-T H.324
Camera:	digital CCD sensor
Display:	local view display-LCD
Audio:	frequency response 50Hz~3.4kHz
Transmission line:	regular telephone line

Block Diagram of H.324 Video Phone



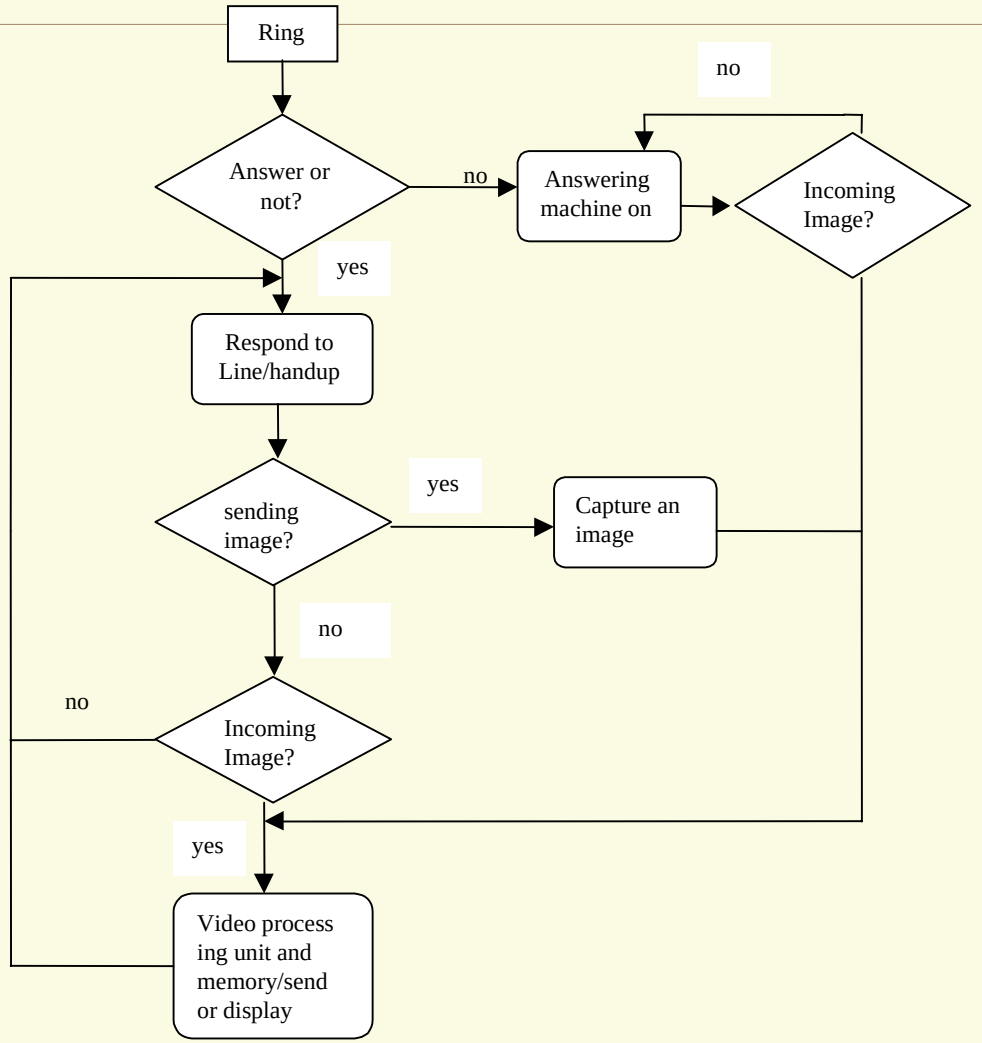
System Elements

- Video I/O equipment including cameras and monitors, their control and selection.
- Audio I/O equipment including microphone and loudspeaker, telephone instrument or equivalent.
- Human user system control, user interface and operation.
- The Video codec (H.263) is for low bit rate communication.
- The Audio codec (G.723) encodes the audio signal from the microphone and decode the audio signal output to speaker.
- The Control protocol (H.245) for multimedia communication which provides capability exchange, signaling of commands and indications, and messages to open and fully describes the contents of logic channels.

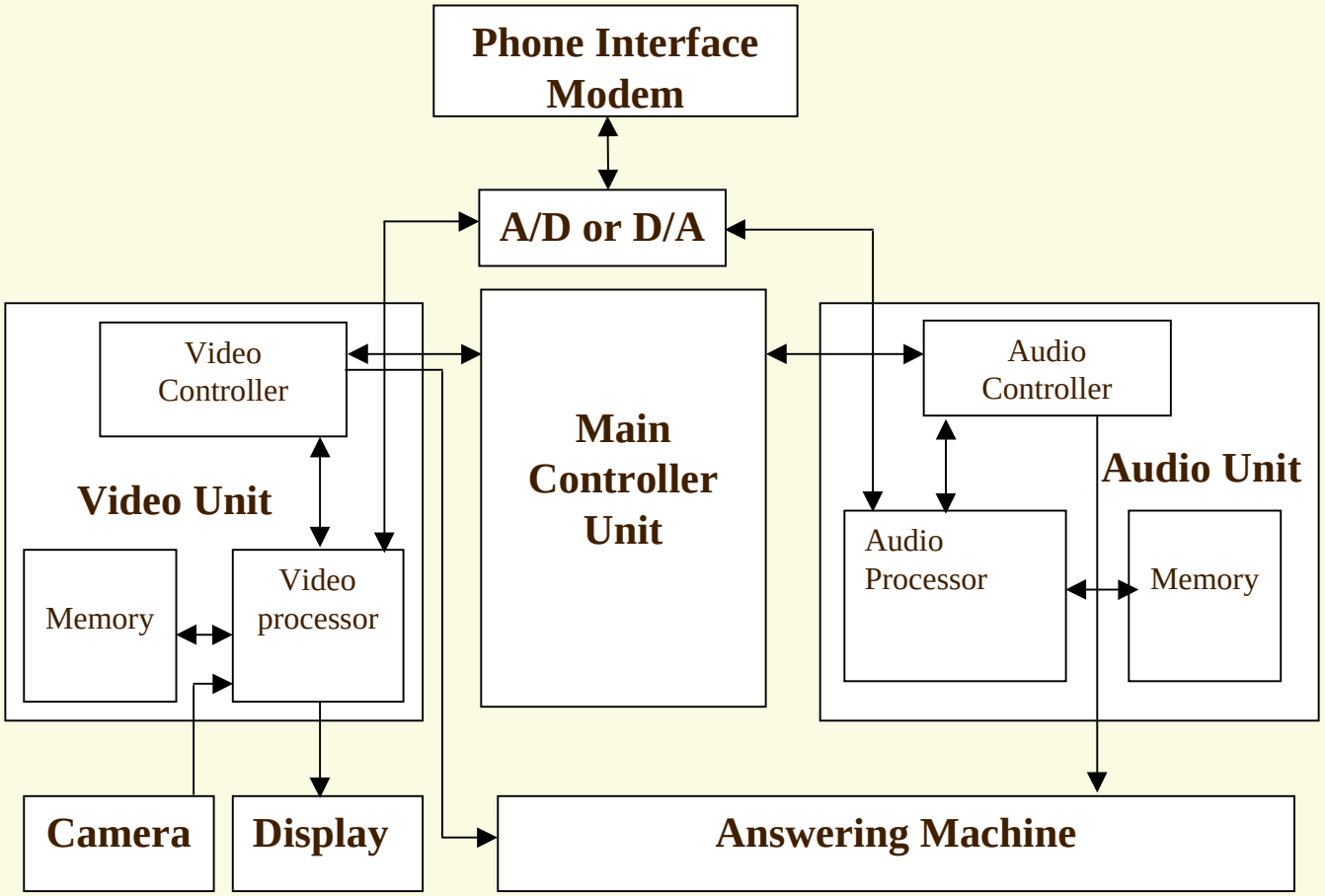
System Elements (cont'd)

- The Multiplex protocol (H.223) is for low bit rate multimedia communication which allows the transfer of any combination of digital voice/audio, digital video/image and data information over a single communication link.
- The Modem (V.34)converts the H.223 synchronous multiplexed bit stream into an analog signal that can be transmitted over GSTN, and converts the received analog signal into a synchronous bit stream that is sent to the Multiplexer/Demultiplexer unit.

Operation Flow Chart



Structure block diagram



Feasibility Study

The videophone system includes: 2 DSP processors, 3 microcontroller, 16M memory, 1 LCD display panel, 1 digital CCD camera, and 1 modem.

Video processor	A processor chipset from Cirrus Logic(CS7615). a low-power analog front-end processor for standard four-color interline transfer CCD imagers. The architecture includes a correlated double sampler, AGC amplifier, black-level clamp, 10-bit ADC, and a complete multi-sync CCD timing generator. The analog CCD imager output can be directly connected to the CS7615 input, which does not require an external buffer amplifier. The pixel data is double-sampled for improved noise performance, and gain adjusted prior to being digitized by the ADC. In large quantities the chip could be under $15.
Audio processor	a 24 bit DSP processor from Motorola (DSP56362) which is available for $5.

Feasibility Study (cont'd)

Memory	for 30 minutes of audio/video stream, 15MB memory is required, which includes 4MB DRAM, 1MB SRAM and 8MB additional memory. The total price is at $20 or so.
Camera	a 320×240 resolution USB camera will be used as built-in camera at about $35.
LCD	the most expensive component in our systems. A LCD 5' TFT can be purchased at $100 for large quantities.
Modem	a V.34 modem is needed in this system which complies with H.324 standard t allow the receive and/or transmit bit rates on the telephone line to be changed without interruption of the data stream. It's easy to find 56k modem costs $15 in the markets, like 56k PCTEL PCI modem.
Microcontroller	general purpose controller from Motorola (MC68HCV1, which is at $7 or so in large quantity.

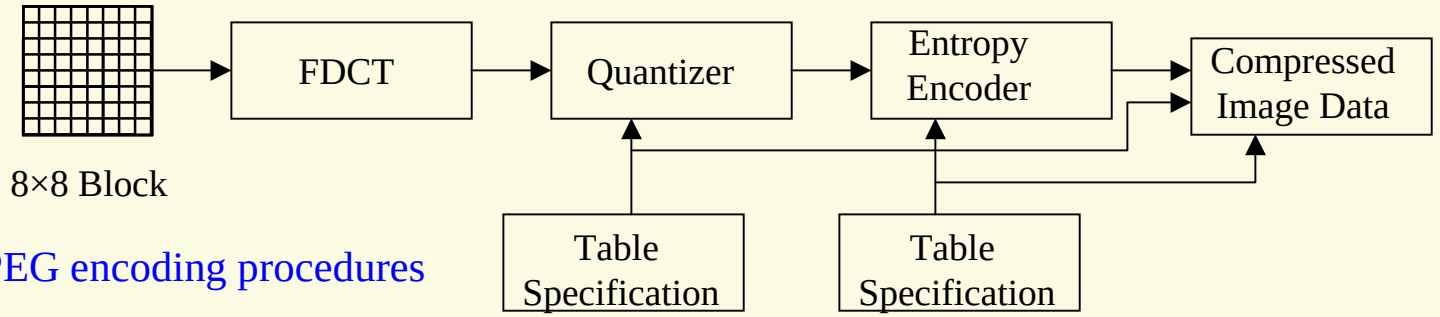
Feasibility Study (cont'd)

Cost Evaluation

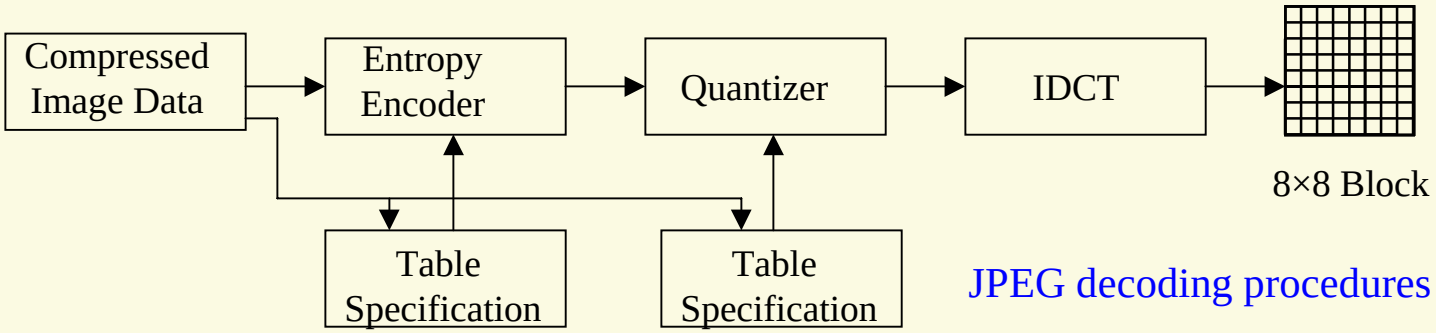
parts:	~$ 225
development:	~$300,000 in equipment labor for 1 years
cost per unit:	~$235 at 10,000 untis
margin profit:	at 30%
sales price:	~$330

Image Processing

JPEG image compression standard has been employed for still image processing. A subset of full JPEG specification, called baseline sequential DCT-based coding is used. The basic sequences of JPEG encoding and decoding procedures are:



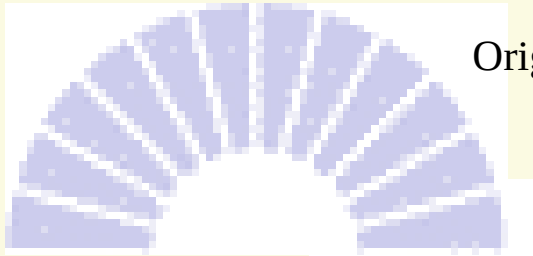
JPEG encoding procedures



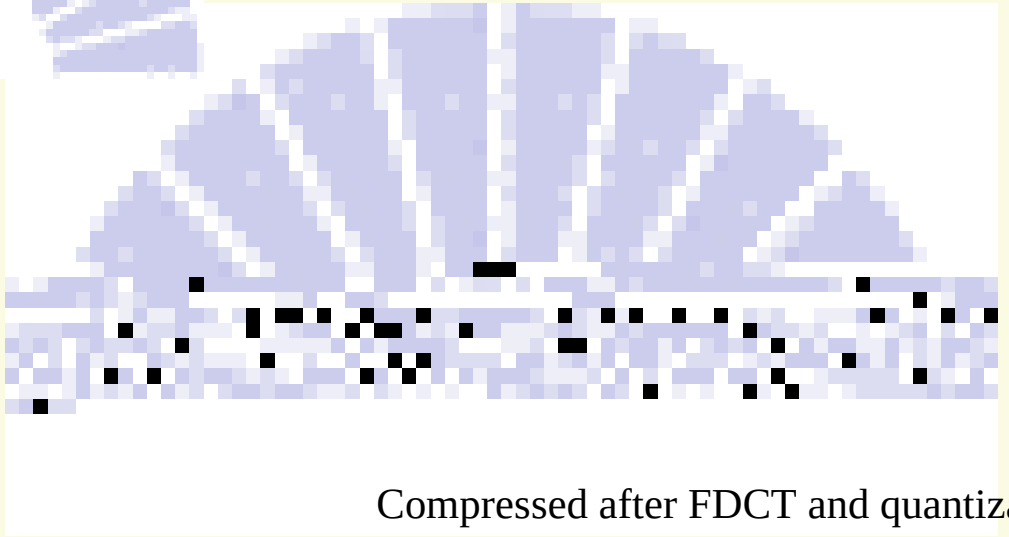
JPEG decoding procedures

Image Processing (cont'd)

An example of image compression and decompression using DCT process. The C++ source code was used to generate the example compressed/decompressed image.



Original image

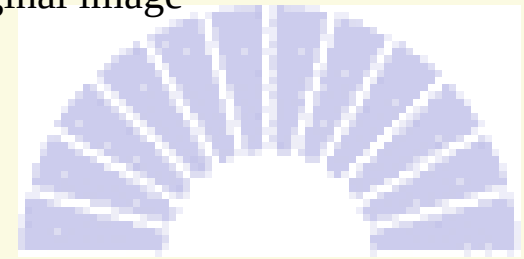


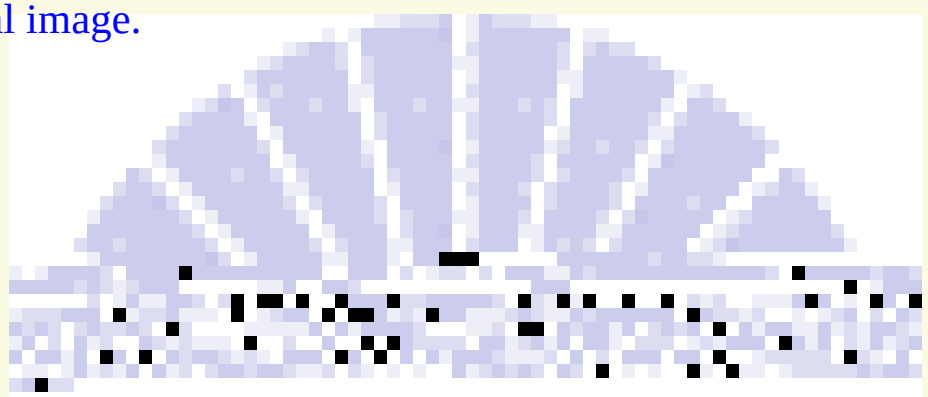
Compressed after FDCT and quantization processing but without entropy encoding

Image Processing (cont'd)

After IDCT processing, the compressed image is tried to be restored but with some loss compared to the original image.

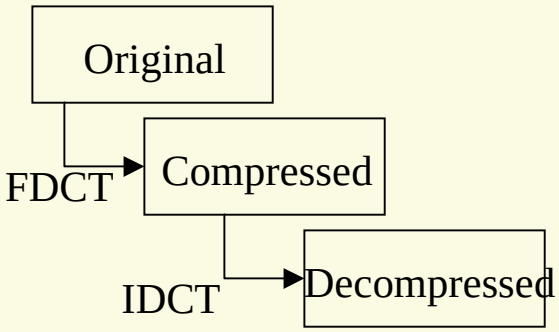
Original image



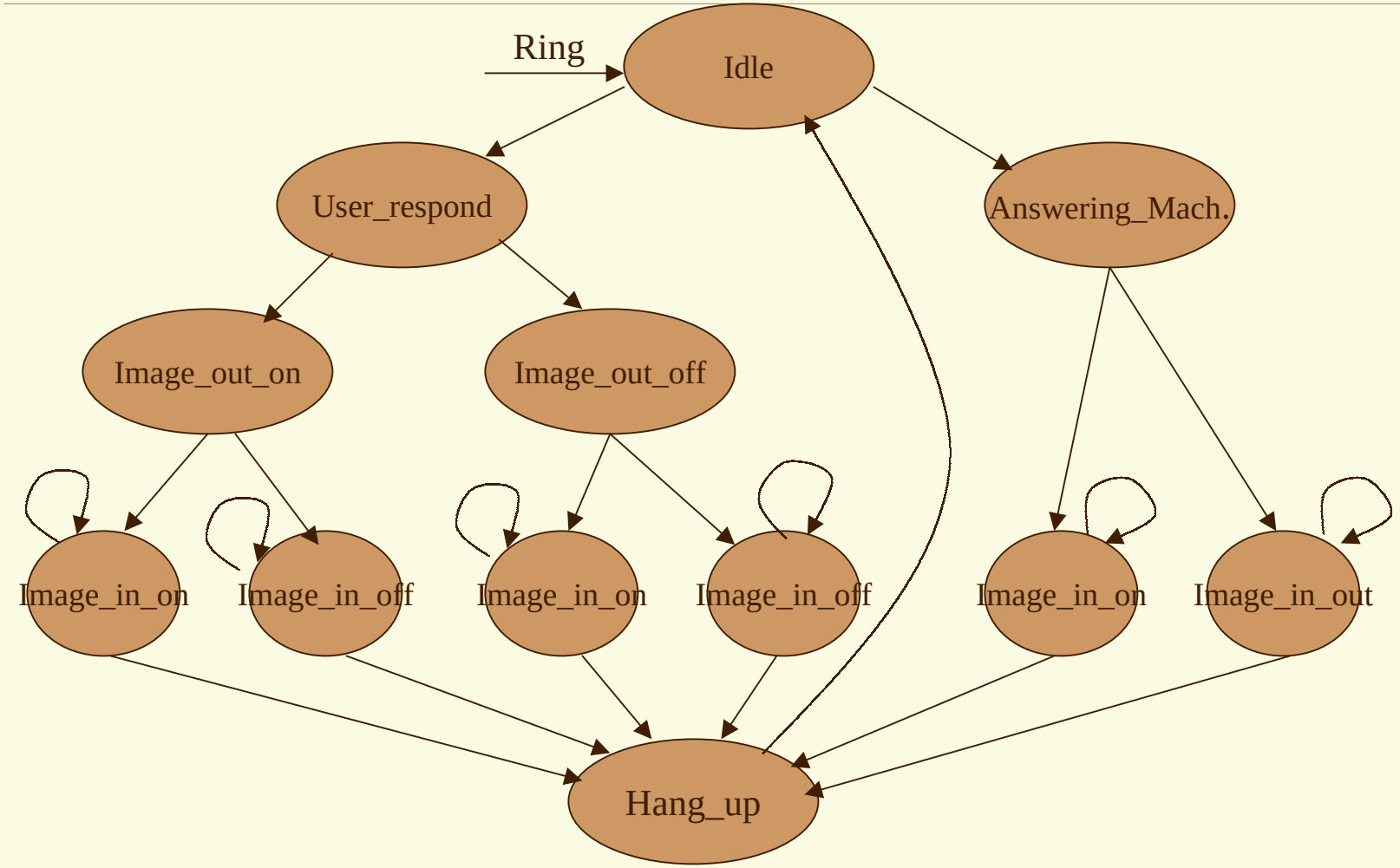


Compressed after FDCT and quantization processing but without entropy encoding

Decompressed after IDCT and quantization processing but without entropy encoding



Finite State Machine (FSM)



VHDL Simulation and Synthesis

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY TRY_4 IS
     PORT (CLK,An,hang_up,IMI,IMO,RESET: IN std_logic;
     audio_rec,image_display,image_send,video_rec : OUT
     std_logic);
END;

ARCHITECTURE BEHAVIOR OF TRY_4 IS
     TYPE type_sreg IS (Hangup,Idle,
     Imagein_on,Imagein_off,Imageout_off,Imageout_on,Respond,A
     ns_machine, Ansin_on, Ansin_off, Iooin_on, Iooin_off);
     SIGNAL sreg, next_sreg : type_sreg;

BEGIN
     PROCESS (CLK, RESET)
     BEGIN
          IF ( RESET='1' ) THEN
               sreg <= Idle;
          ELSIF CLK='1' AND CLK'event THEN
               sreg <= next_sreg;
          END IF;
     END PROCESS;

     PROCESS (sreg,An,hang_up,IMI,IMO)
     BEGIN
          audio_rec <= '0'; image_display <= '0'; image_send
     <='0';video_rec <= '0';
          next_sreg<=Hangup;

          CASE sreg IS
               WHEN Hangup =>
                    video_rec<='0';
                    audio_rec<='0';
                    image_send<='0';
                    image_display<='0';
                    next_sreg<=Idle;
```

VHDL Simulation and Synthesis (cont'd)

```
WHEN Idle =>
     video_rec<='0';
     audio_rec<='0';
     image_send<='0';
     image_display<='0';
     IF ( An='1' ) THEN
          next_sreg<=Respond;
     ELSE
          next_sreg<=Ans_machine;
     END IF;

WHEN Respond =>
     video_rec<='0';
     audio_rec<='0';
     image_send<='0';
     image_display<='0';
     IF ( IMO='1' ) THEN
          next_sreg<=Imageout_on;
     ELSE
          next_sreg<=Imageout_off;
     END IF;

 WHEN Imageout_on =>
      video_rec<='0';
      audio_rec<='0';
      image_send<='1';
      image_display<='1';
      IF ( IMI='1' ) THEN
           next_sreg<=Imagein_on;
      ELSE
           next_sreg<=Imagein_off;
      END IF;
```

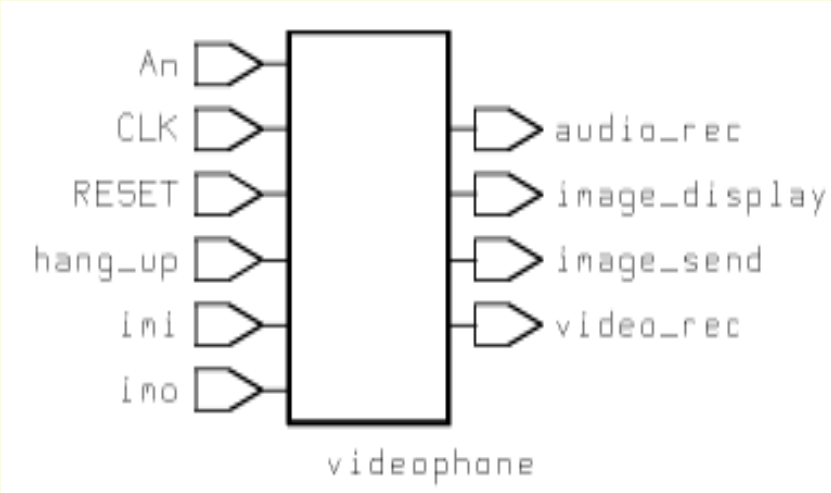
```
WHEN Imageout_off =>
     video_rec<='0';
     audio_rec<='0';
     image_send<='0';
     image_display<='0';
     IF ( IMI='1' ) THEN
          next_sreg<=Iooin_on;
     ELSE
          next_sreg<=Iooin_off;
     END IF;

WHEN Imagein_on =>
     video_rec<='1';
     audio_rec<='0';
     image_send<='0';
     image_display<='1';
     IF ( hang_up='1' ) THEN
          next_sreg<=Hangup;
     ELSE
          next_sreg<=Imagein_on;
     END IF;
```

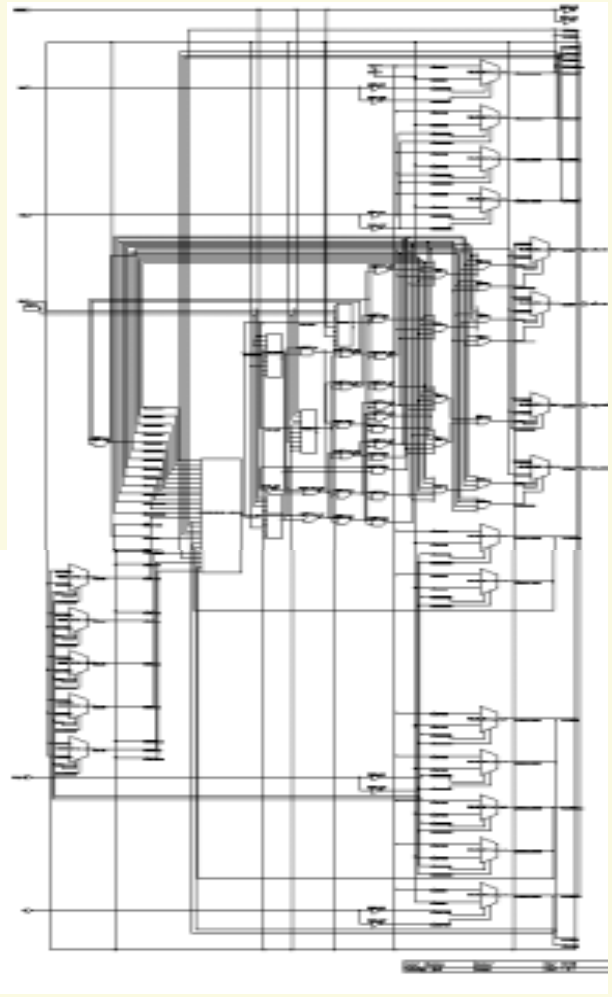
VHDL Simulation and Synthesis (cont'd)

```
            next_sreg<=Hangup;
      ELSE
            next_sreg<=Imagein_off;
      END IF;

WHEN Iooin_on =>
      video_rec<='1';
      audio_rec<='0';
      image_send<='0';
      image_display<='1';
      IF ( hang_up='1' ) THEN
            next_sreg<=Hangup;
      ELSE
            next_sreg<=Iooin_on;
      END IF;

WHEN Iooin_off =>
      video_rec<='0';
      audio_rec<='0';
      image_send<='0';
      image_display<='0';
      IF ( hang_up='1' ) THEN
            next_sreg<=Hangup;
      ELSE
            next_sreg<=Iooin_off;
      END IF;

WHEN Ans_machine =>
      video_rec<='0';
      audio_rec<='1';
      image_send<='0';
      image_display<='0';
      IF ( IMI='1' ) THEN
            next_sreg<=Ansin_on;
      ELSE
            next_sreg<=Ansin_off;
      END IF;

WHEN Ansin_on =>
      video_rec<='1';
      audio_rec<='1';
      image_send<='0';
      image_display<='1';
      IF ( hang_up='1' ) THEN
            next_sreg<=Hangup;
      ELSE
            next_sreg<=Ansin_on;
      END IF;

WHEN Ansin_off =>
      video_rec<='0';
      audio_rec<='1';
      image_send<='0';
      image_display<='0';
      IF ( hang_up='1' ) THEN
            next_sreg<=Hangup;
      ELSE
            next_sreg<=Ansin_off;
      END IF;

      WHEN OTHERS =>
   END CASE;
END PROCESS;
END BEHAVIOR;
```

VHDL Simulation and Synthesis (cont'd)

Gate level system synthesis generated from VHDL code





Conclusions

- A videophone system has been designed based on ITU compatible standard.
- Operation function, feasibility, and structure diagram have been provided and analyzed.
- An example focused on baseline sequential FDCT and IDCT based coding for image compression and decompression is to demonstrate the typical JPEG image processing.
- VHDL simulation and synthesis has been made to generate a gate level structure to fulfill the required function.

References

1. V. Bhaskaran, K.Konstantinides, *Image and video compression standard Algorithms*, Kluwer Academic Publishers

2. Phillip E. Mattison, *Practical digital video with programming examples in C*, John Wiley & Sons, Inc. 1994

3. S. Yalamanchili, *VHDL Starter's Guide*, Prentice Hall, 1998

4. Intel® PC Cameras, Technical Guide to Video Phone Calls, http://developer.intel.com/pccamera/white.htm

5. Crystal ®, CS7615: 10-Bit A/D Programmable Timing generator for Interlaced CCDs. http://www.cirrus.com/products/overviews/cs7615.html