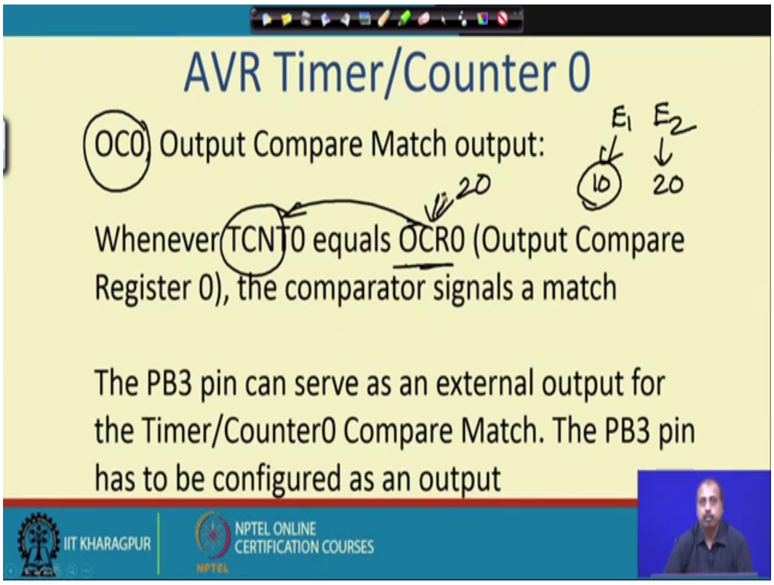
Microprocessors and Microcontrollers
Prof. Santanu Chattopadhyay
Department of Electrical & Electronics and Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture-52
AVR (Contd.)

So, the AVR time are counted 0. So, it has got its IB timer and also if the clock source is external it will act as a counter and the many very important feature that it has is that, there is an output compared match. So, basically when the timer is operating or the counter is operating so you can you can have some intermediary values at which you want to determine a match.

(Refer Slide Time: 00:44)

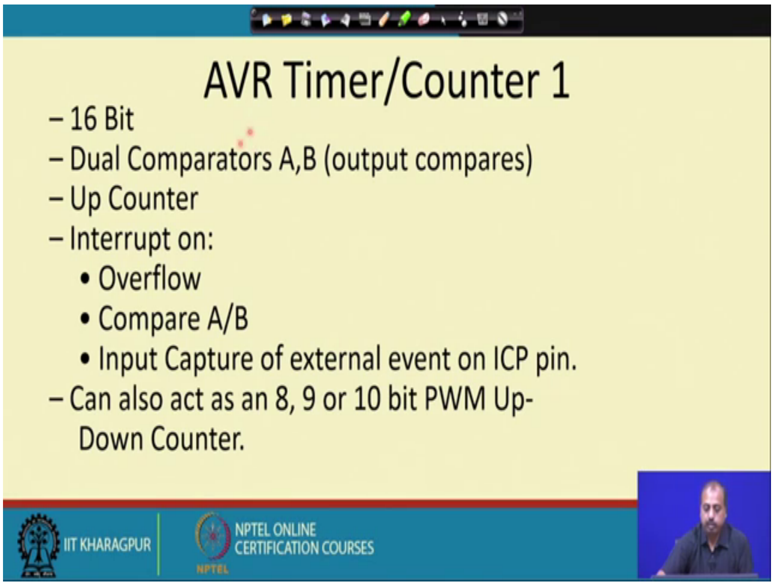


For example if I have got say 2 activities E 1 and E 2 to be E 1 and E 2 scheduled. So, this and what we want is that this E 1 and E 2 they should be created say this one after say 10, 10 seconds and this one after 20 second. So, in normal 8051 type of operation so, what you have to do is you have to use 2 different timers and then one for 10 second and another for 20 second and accordingly generate 2 events and get the overflows detected and schedule those events.

However in case of AVR microcontroller so this feature is there where we have got another register which is called output compare match, output compare register OCR. So, in this OCR. So, you can load this values say 10 initially. So, that and start the timer. So, this timer we need to goes up to 10. So, it will this TCNT 0 this will match with this OCR 0 and accordingly this OC 0 output. So, this output will be made one. So, you get that information that this is the time for activate, activate the action E 1 after and then you can reload this OCR 0 with the value 20.

So, that again when the count reaches 20 so you will get another match and then you can start the activity E 2. So, this way it this is useful because I can have multiple activity started at different times which are controlled by the same timer ok. So, that is the feature with this TCNT this TCNT at 0 registered and this OCR 0 registered.

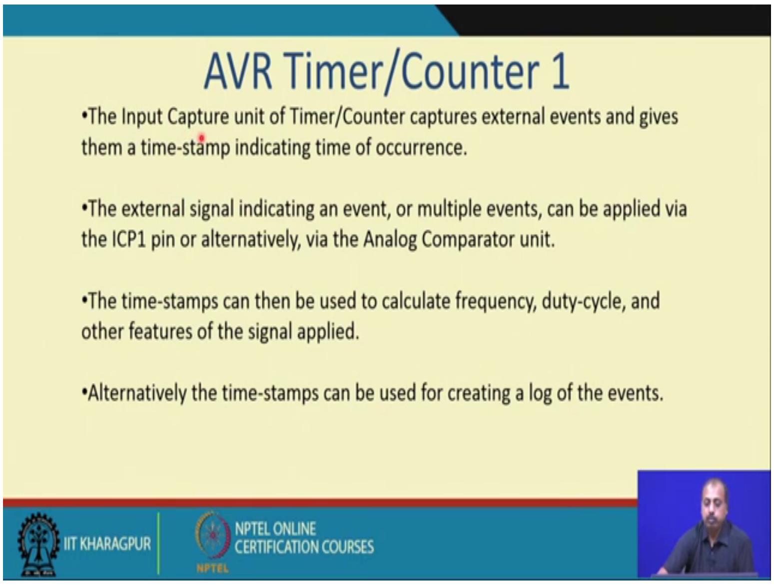
(Refer Slide Time: 02:32)



Another timer that we have is a 16 bit timer; those 16 bit timer is that a timer counter 1. So, 16 bit timer it has got dual comparators A and B there are 2 output comparators A and B again this are up counter. So, so we have been previous example we have got only single comparator ok. So, OCR 0 was there, but here I have got 2 comparators A and B.

So, otherwise it is same it is up counter. So, interrupt on overflow that feature is there comparison with A and B here OCR registered so that is there and this input capture of external events is on ICP pin. So, ICP pin. So, you can capture this external event. So, accordingly you can load it, it can also act as pulse width modulation facilities can be provided so, 8, 9 or 10 bit pulse width modulation with up down counter. So, this AVR this counter one can also be moduled as PWM, so we will see that slowly.

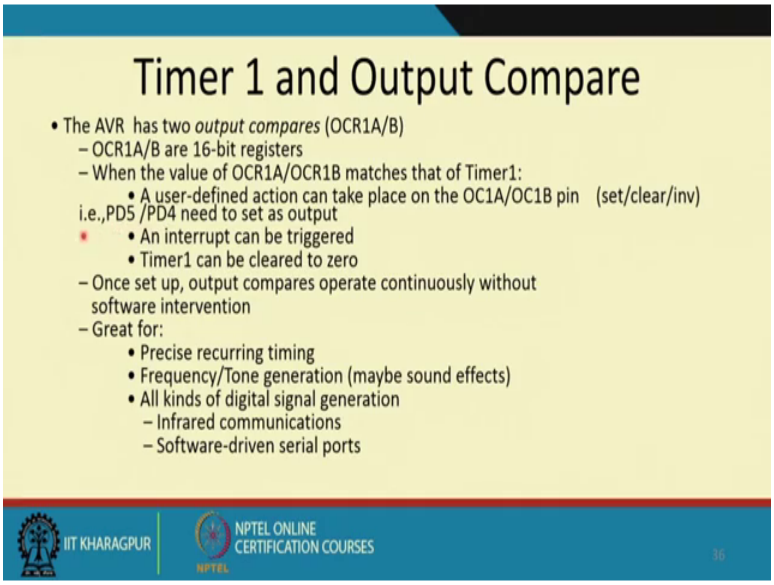
(Refer Slide Time: 03:34)



Then this input capture unit captures the external events and gives them a time stamp indicating their time of occurrence. So, when that external event has occurred. So, that time will be captured there, the external event signal indicating an event or multiple events can be applied by the ICP one pin or it can be via this ad convertor analog to digital convertor. So, the analog comparator module can be used for during that purpose, this time stamp we can note down the, we can, we can use it for calculating frequency duty cycle.

So, as I was telling you in the last class that you can we can capture 2 different points of signal and accordingly you can tell, you can determine say the frequency of the signal or you can determine the duty cycle of the signal. So, those type of application can be done, it can also be used for creating a log of events, like when say if I say that I am counting the items coming on conveyor belt and I also want to take a note like we when individual items actually came to the point. So, that way I can do this by noting down the corresponding time stamps.

(Refer Slide Time: 04:44)

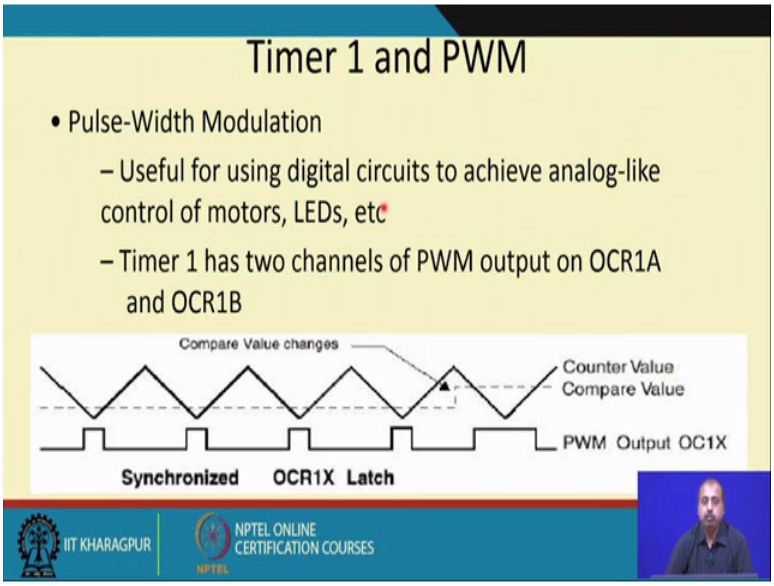


So, AVR 1 timer one it has got 2 compare outputs OCR 1 A and OCR 1 B. So, they are 16 bit registers because the timer itself, timer one itself is 16 bit. So, these are also 16 bit registers the value of OCR 1 and OCR B matches with that of timer one user defined action will take place on this OCA, OC 1 A and OC 1 B pin. So, the actions are set clear or invert. So, we can programme it like this that when this match occurs, so the particular pin will be set it will be cleared or it will be inverted.

So, or another activity can be an interrupt can be made to generate. So, you can tell the processor that when this timer comes to this particular set value in the set in the OC register. So, you can it will trigger an interrupt. So, or the timer 1 can be cleared to 0. So, that is also another activity once set up this output compares output continually without software intervention. So, you just programme it once and from that point onwards it continues, it it is good for precise recurring timing.

Like you have to do some event in a periodic way after some periodic intervals of time so you can do this operation so, you can take the help of this timer 1 and load this OCR 1 A or 1 B register accordingly. So, that at periodic intervals, so you get an indication that this is the time to do that periodic operation, frequency or tone generation. So, different sounds you want to generate various points of times. So, that can be done and digital signal generation for infrared communication software driven serial ports. So, that type of application also. So, you can generate the control signal. So, particularly when we have to develop some hand shaking signals at periodic intervals or at some fixed time points. So, you can do that using these timers.

(Refer Slide Time: 06:43)



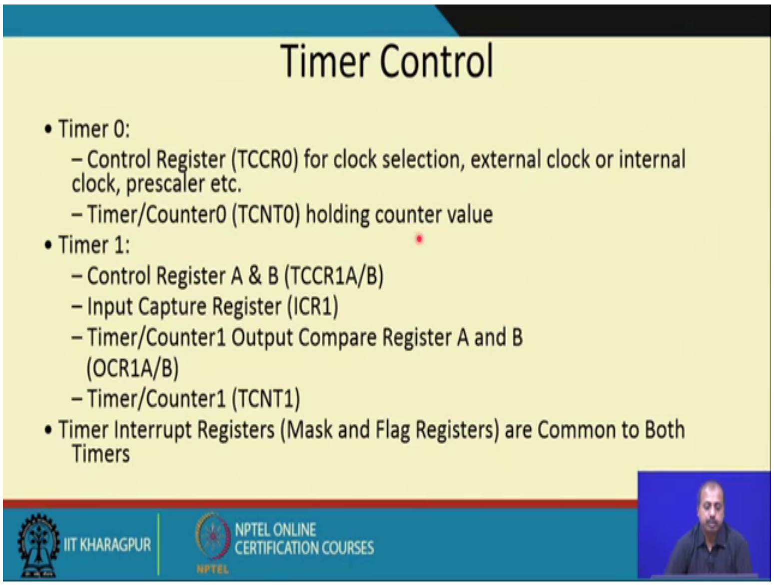
So, timer one can also be used for pulse-width modulation. So, what is pulse pulse-width modulation? So, this is the diagram that shows how is it occurring, you see that suppose this is a triangular width and there are 2 values, one is the low value of the of the pulse it is basically, so we set a value. So, when you when this signal goes below this point when this, this triangular wave goes below this set value we generate a high pulse. So, this signal makes a transition from low to high, at this point the signal is going below the set value. So, it is generating it is becoming high.

Now, after some time the signal will again rise and it crosses the say crosses the triangular the set value again. So, at this point the signal goes low. So, you see that if I set that this as the comparator value then whenever signal is high higher than this the value of the output is 0 and whenever the this triangular wave value is less then this set value the value of the output is 1.

So, you get a pulse like this now if you change this dc level, if you change this dc level. So, you can have different pulse width like for example, here suppose at this point we have changed the dc level from this one to this one. So, when we change this accordingly. So, that cut off will occur. So, at this low to high transition will occur when as when the pulse is at the when the triangular wave is at this point and similarly this will go low when the triangular wave is at this point. So, this way we can have this pulse width modulated output.

So, this is useful for driving motors or say glowing led. So, if you have to generate many times communication also this pulse width modulation is necessary. So, whenever you need these sort of things. So, you can, you can change the carrier signal feature and so based on the base signal. So, here the dotted line is the base signal. So, those changes the carrier signal feature when the carrier is crossing this they are generating a pulse of larger, when the base value is the compare value is higher. So, this is it has got 2 channels of this pulse width modulation output or OCR 1 A and OCR 1 b. So, that is useful for many applications. So, this particular facility we do not have in other micro controllers like say 8051. .

(Refer Slide Time: 09:17)

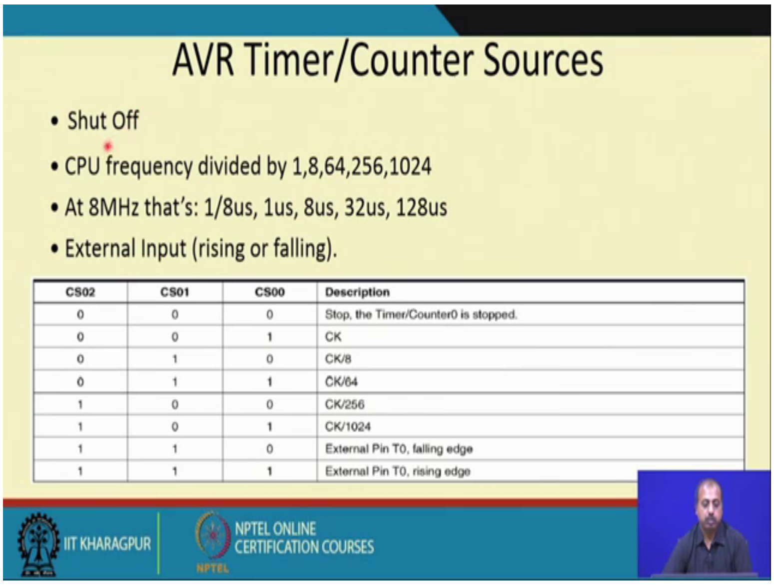


So, timing control, so you have seen that there is a timer 0 register, there is a timer 1 register. So, timer 0 register has got a control register TCCR 0 for clock selection which is external clock or internal clock then there prescaler that is from which point it will start, then timer counter 0. So, this the this will hold the counter value. So, these are the 2 registers relevant for timer 0.

For timer one we have got control register TCCR 1 A and 1 B, there are 2 control registers similarly we have got 2 count. So, count value is there TCNT 1, so this is the count value. So, you can have this input capture register ICR 1 where the capturing input at some point, then timer counter one output compare register A and B. So, this is OCR register was there for timer 0 here also we have got OCR 1 A and OCR 1 B. So, here we have got another register in timer 0 which is OCR 0.

So, that is not written here explicitly. So, that is another register we had for this output comparator, output comparison and then timer interrupt registers are there. So, both the timer they have got mask and flag register. So, if they are not detailed here, but you can mask those re register you can out a mask for those for the corresponding interrupts ok. So, that can be that can be useful just like in 8051 we have seen here also we can do that thing.

(Refer Slide Time: 10:46)

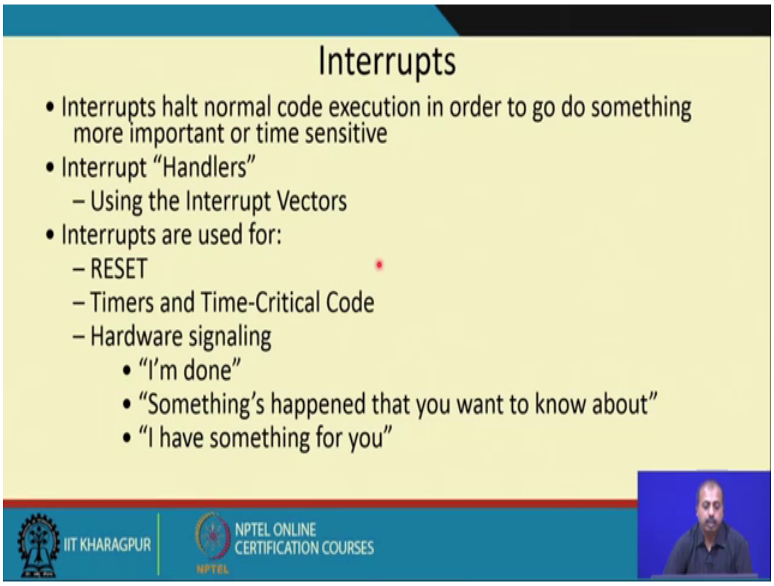


CS02	CS01	CS00	Description
0	0	0	Stop, the Timer/Counter0 is stopped.
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T0, falling edge
1	1	1	External Pin T0, rising edge

So, this timer counter sources one is the shut off mode where nothing has to be done like this, this are the control lines CS02, CS01, CS00. So, if these bits are 000; that means, it is stop and the timer counter will be is a stop. So, if the value is 001 then the operation will be at the normal clock of the microcontroller. So, the timer counter will operate at the normal clock of the micro controller, then 010 it will operate at clock by 8 frequency 011 will be clock by 64 facility frequency. So, this way up to you see there are good number of setting, 8, 64, 256 and 1024 and of course, we have got default 1 setting.

So, this CPU frequency may be divided by these values and accordingly it will be at 8 so at 8 mega hertz. So, these are one eighth micro second 1 micro second 8 micro second. So, like that. So, you can change the clock that is fit to the counter, timer counter by setting this CS 0 bits and we can also have this external source like say, 110 and 111 these are for the external pin t 0 one is falling edge another is rising edge. So, if you are if you are taking it from the outside world the clock frequency is coming from outside world. So, this can be this can be fate to the pin t 0 and that will be either falling edge triggered or rising edge triggered.

(Refer Slide Time: 12:19)



So, if you look into the interrupts the interrupt handlers so it will halt for any interrupt this is true that any halt normal code execution for doing something sensitive. So, we have got this interrupt handler. So, that interrupt service routine so here the name is interrupt handler for interrupts vectors we have, then the interruptions we use for resetting then timers time critical code like that or hardware signalling like something happens in the hardware. So, that way you can signal that. So, there are external pins for feeding that interruption just like other interrupts. So, we other processors we have this thing.

(Refer Slide Time: 12:54)

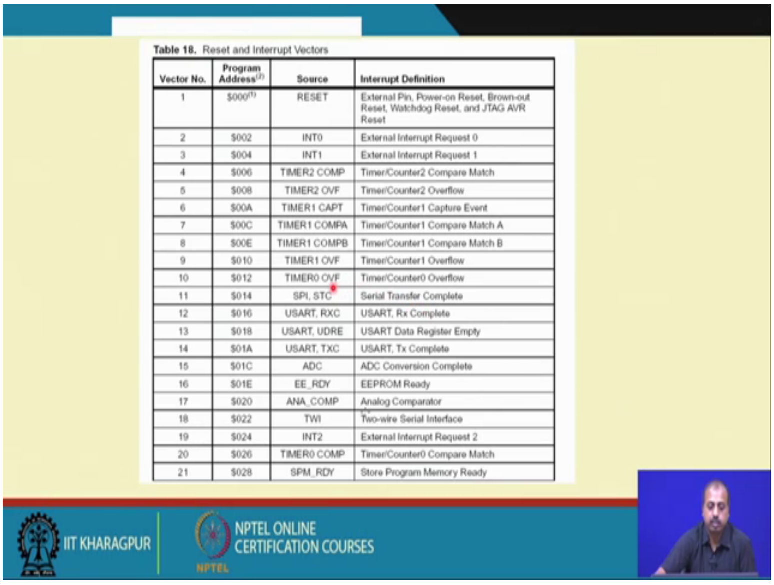


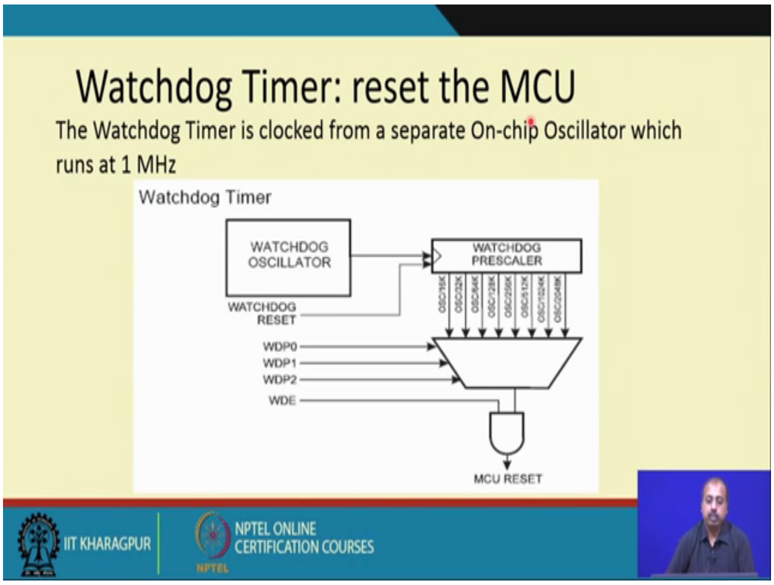
Table 18. Reset and Interrupt Vectors

Vector No.	Program Address[2]	Source	Interrupt Definition
1	$000[1]	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$002	INT0	External Interrupt Request 0
3	$004	INT1	External Interrupt Request 1
4	$006	TIMER2 COMP	Timer/Counter2 Compare Match
5	$008	TIMER2 OVF	Timer/Counter2 Overflow
6	$00A	TIMER1 CAPT	Timer/Counter1 Capture Event
7	$00C	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	$00E	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	$010	TIMER1 OVF	Timer/Counter1 Overflow
10	$012	TIMER0 OVF	Timer/Counter0 Overflow
11	$014	SPI, STC	Serial Transfer Complete
12	$016	USART, RXC	USART, Rx Complete
13	$018	USART, UDRE	USART Data Register Empty
14	$01A	USART, TXC	USART, Tx Complete
15	$01C	ADC	ADC Conversion Complete
16	$01E	EE_RDY	EEPROM Ready
17	$020	ANA_COMP	Analog Comparator
18	$022	TWI	Two-wire Serial Interface
19	$024	INT2	External Interrupt Request 2
20	$026	TIMER0 COMP	Timer/Counter0 Compare Match
21	$028	SPM_RDY	Store Program Memory Ready

So, this is the table that contains this interrupt vectors. So, vector number 0 is the reset, so this is external pin poweron, reset, brownout reset. So, these are all they will come to this reset there are. So, many points by which this reset can come then INT 0 and INT 1 these are 2 external interrupts ok. So, their vector numbers are 2 and 3, then this is timer intercounter 2 compare match, timer counter 2 overflow counter timer counter 1 compare match. So, this way we have got all these timer counters overflow and the match interrupts ok. So, these are there.

Then there are serial transfer complete spi stc type of thing then uart their interrupt. So, adc there is an interrupt. So, this way we can have a number of interrupt sources and in case of this AVR micro-controller. So, here it is there are 21 such vector location from which the, at which the interrupts can land and each location you see it has got 2 by. So, 000 and 001 so, they will hold the vector address for the reset. So, there should be the jump instruction that will be 2 bite wide and that will hold the actual ISR entries just like other processors. So, everywhere you see that there is a gap of 2 bites for holding the interrupt service routine address then we have the watchdog timer.

(Refer Slide Time: 14:23)



So, this is the one watchdog timer it will be clocked from a separate onchip oscillator which runs at 1 mega halts it is. So, we can. So, watchdog we know that if any of these events occur then it will send this MCU reset ok. So, so this watchdog timer it overflows in that case it will be it will be coming to this this this reset this output will be 1 ok. So, this when the output is 1 so, it will be sending a reset to the processor.

So, this way this watchdog timer is useful when we have got some activity which was suppose to be complete within some time, but it is not. So, in that case in those cases we can have the, this watchdog timers for doing this operation. So, this way this AVR microcontroller has got many interesting features like say compare to 8051 or even arm. So, we have got many interesting features and they are that is why they are also quite a lot of usage you can find in different imbibed applications.