Motor Control PWM Module

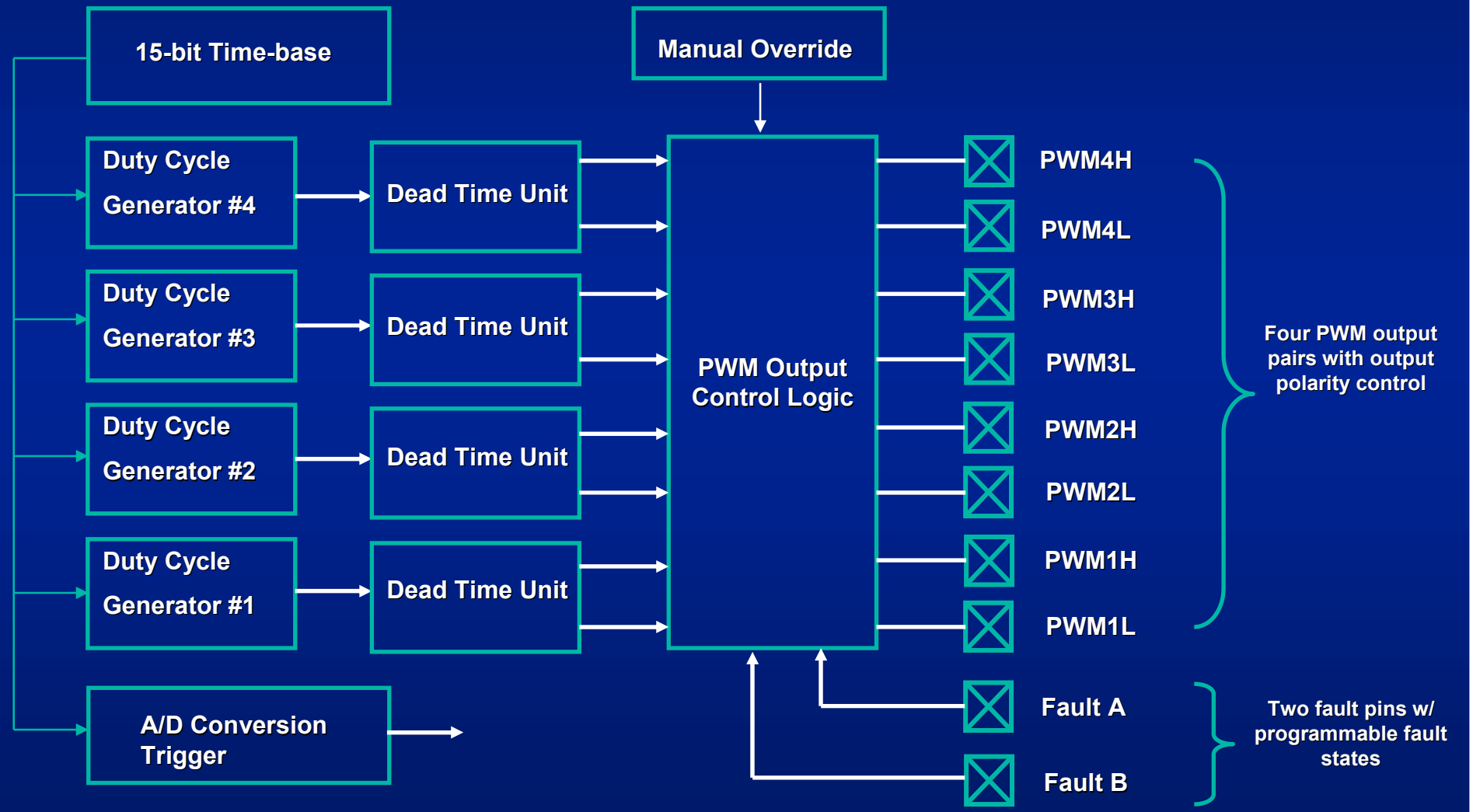
Session Agenda

- MC Peripheral Overview
  - PWM Time Base and Period Setting
  - Duty Cycle Programming
  - Dead Band Generators
  - Output Override
  - Fault pins
  - A/D Converter Synchronization
  - MC PWM in Power Safe Modes

Motor Control PWM

- MCPWM Features
  - Designed for ACIM, BLDC, SR, UPS
  - Dedicated time base with TCY/2 edge resolution
  - Complementary PWM signals w/ dead time
  - Output pin polarity set with configuration bits
  - Special PWM generation modes
  - Multiple output modes
  - Override register for commutation applications
  - Special event trigger for A/D conversion
  - Fault pins for protection of power circuits

MC PWM Block Diagram



MC PWM Registers

- PTCON : PWM Time Base Control register
- PTMR : PWM Time Base counter
- PTPER : PWM Time Base Period register
- SEVTCMP : PWM Special Event Compare register
- PWMCON1 : PWM Control register #1
- PWMCON2 : PWM Control register #2
- DTCON1: Dead Time Control register #1
- DTCON2: Dead Time Control register #2
- FLTACON : Fault A Control register
- FLTBCON : Fault B Control register
- PDCx : PWM Duty Cycle register #1 ~ #4

MC-PWM Timebase

PTCON : PWM Time Base Control Register

- Dedicated 15-bit time-base, period register
  - Up count (edge align), up/down count (center align)
- PWM resolution : Tcy/2 For Example:
  - At 20 MIPS: 19.5 KHz @ 11 bit

    i.e. 11 bit resolution above audible frequencies
  - PWM interrupt every 1-16 periods
- 1:1, 1:4, 1:16, or 1:64 input clock prescaler options
- 1:1 ~ 1:16 Interrupt Postscaler options

Motor Control PWM

- PTCON register
  - PTMOD - Timebase Mode Select Bits
  - PTCKPS - Timebase clock pre-scale
  - PTOPS - Timebase interrupt post-scale
  - PTSIDL - Stop in IDLE mode
  - PTEN - Timebase Enable

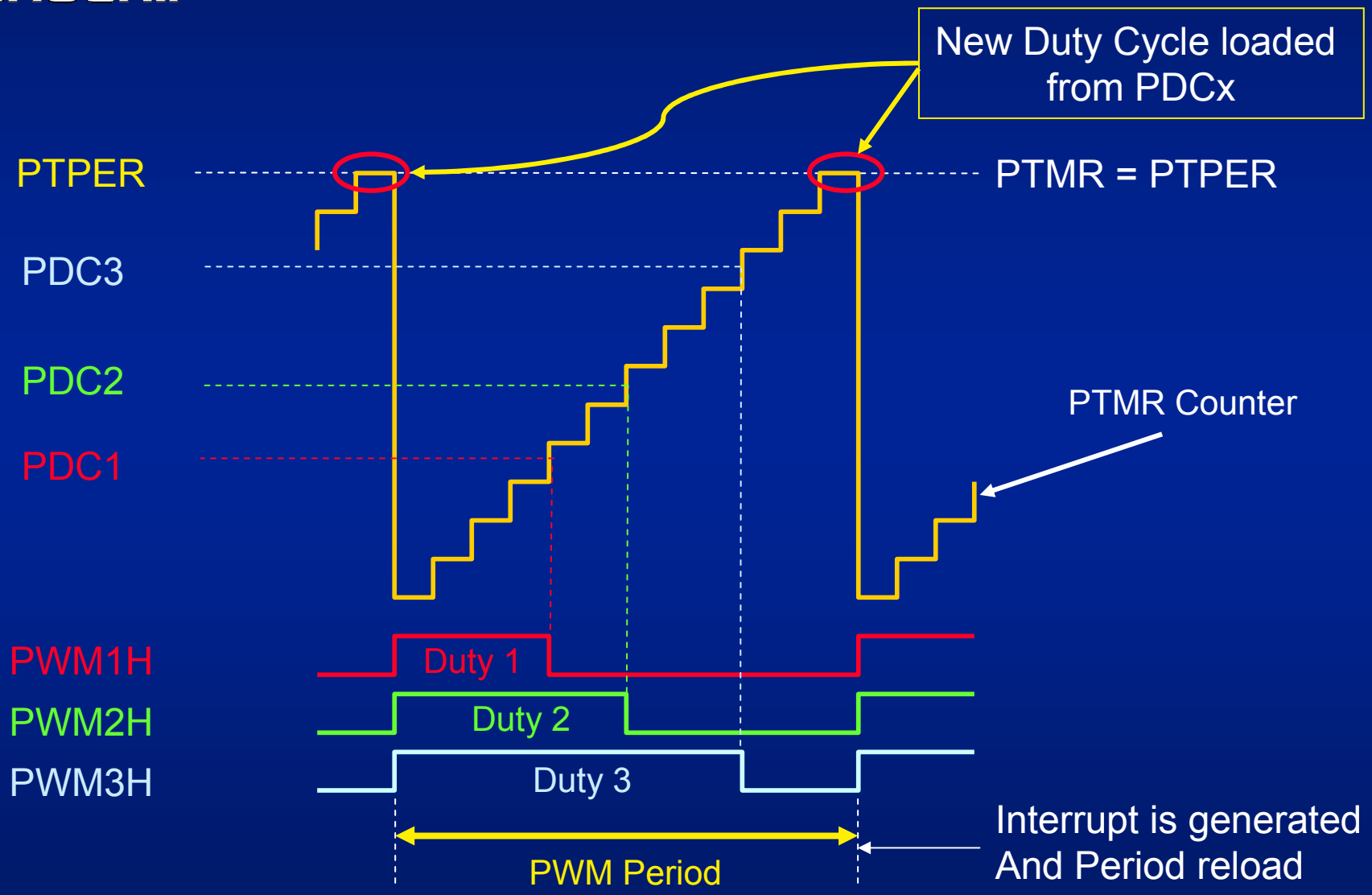
R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
PTEN	-	PTSIDL	-	-	-	-	-
bit15	14	13	12	11	10	9	bit8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTOPS<3:0>				PTCKPS<1:0>		PTMOD<1:0>	
bit7	6	5	4	3	2	1	bit0

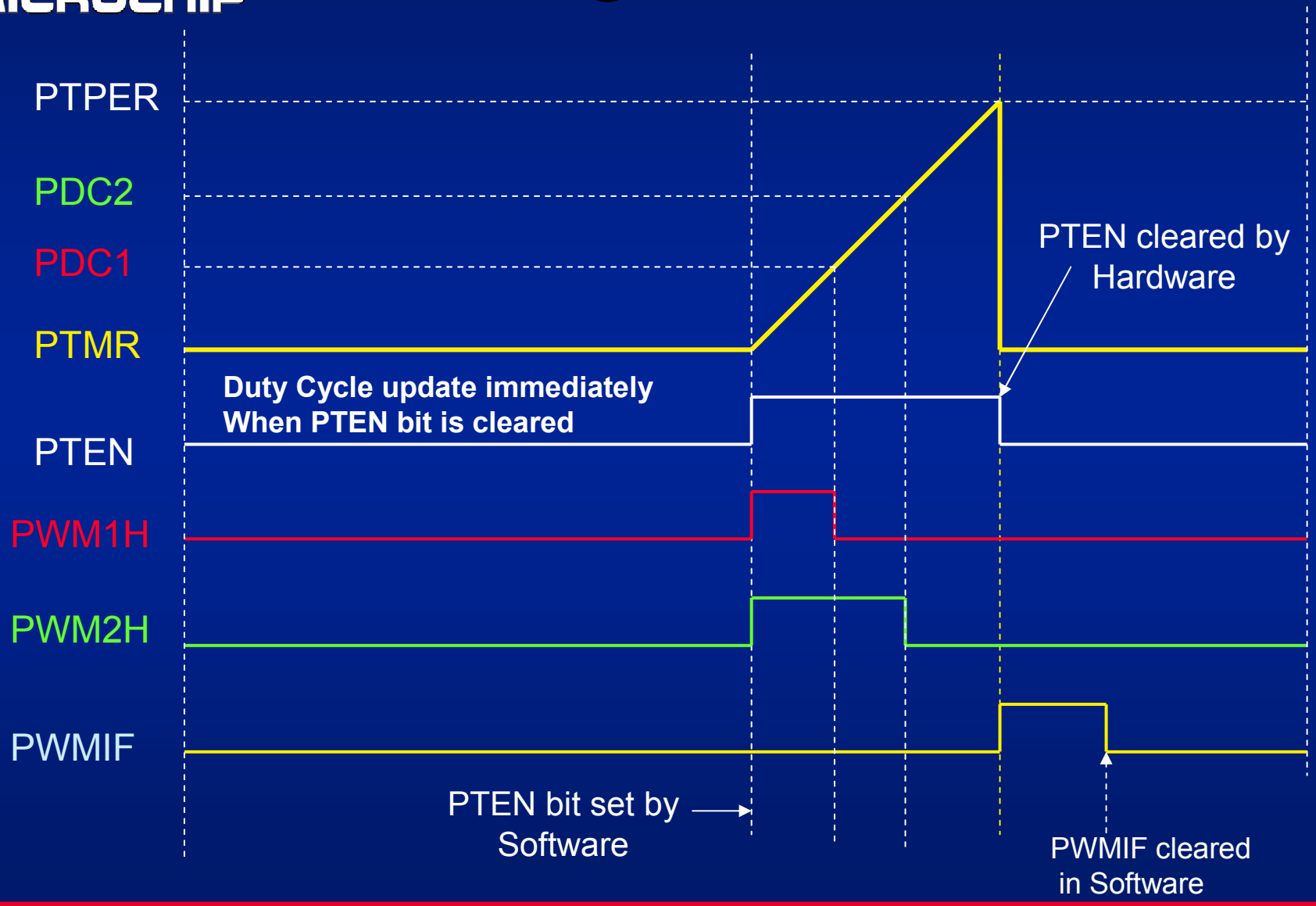
MC PWM Mode

- PTMOD<1:0> bit in the PTCON register
  - = 00 : Free run mode.
    - The PTMR will count upwards until the value is matched with PTPER, PTMR will reset.
  - = 01 : Signal-Event Mode
    - The PTMR will count upwards when the PTEN is set. When the PTMR is matched the PTPER, the PTMR will be reset and also clear the PTEN bit to halt the time Base operation.
  - =10 : Up/Down Counting Mode
    - The PTMR will count upwards then count downwards when the PTMR is matched with PTPER. The PTMR count down to zero then PTMR will change to up-count.
  - =11 : Up/Down Counting Mode with Interrupt for double PWM updates

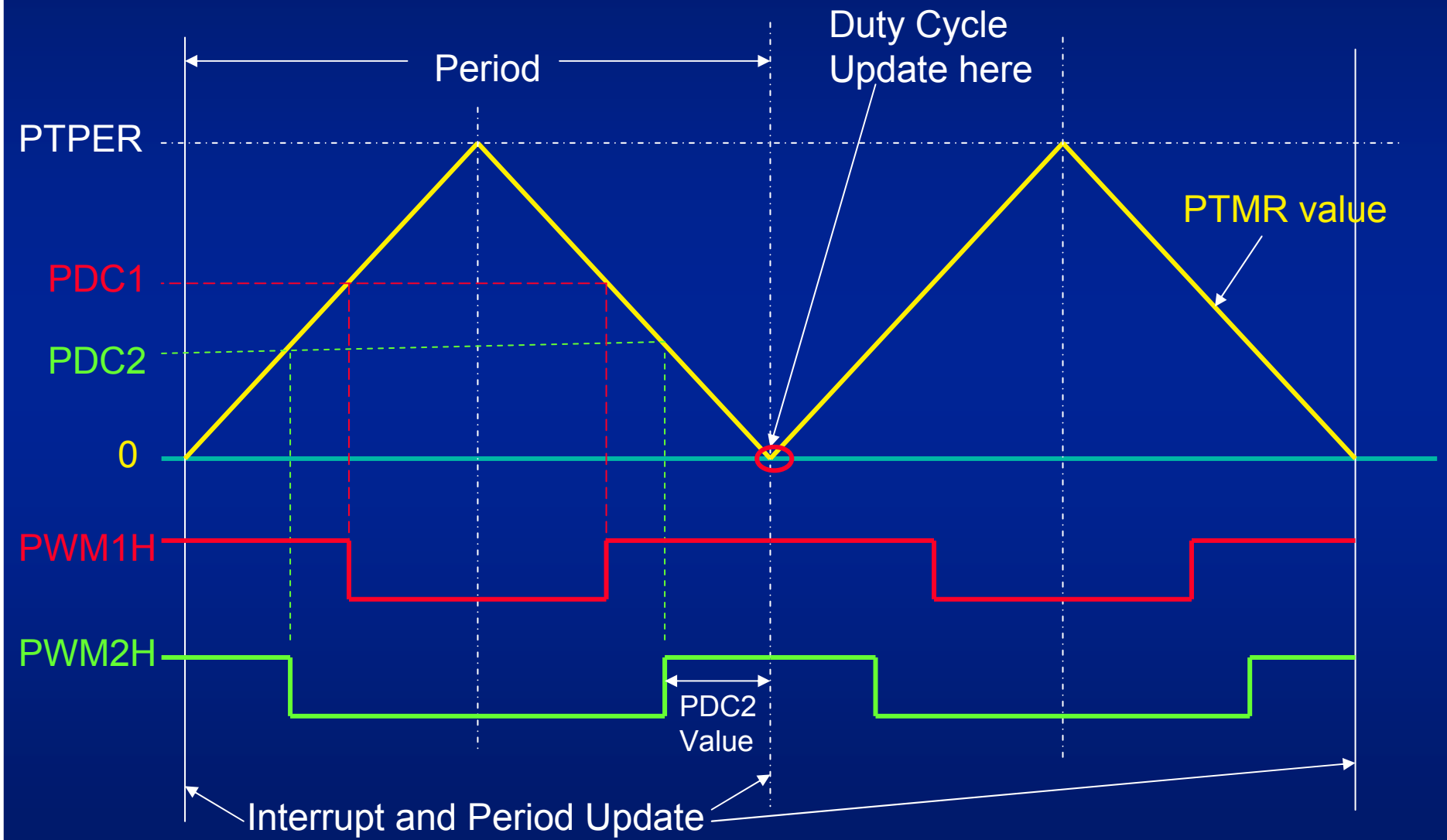
Edge Aligned PWM



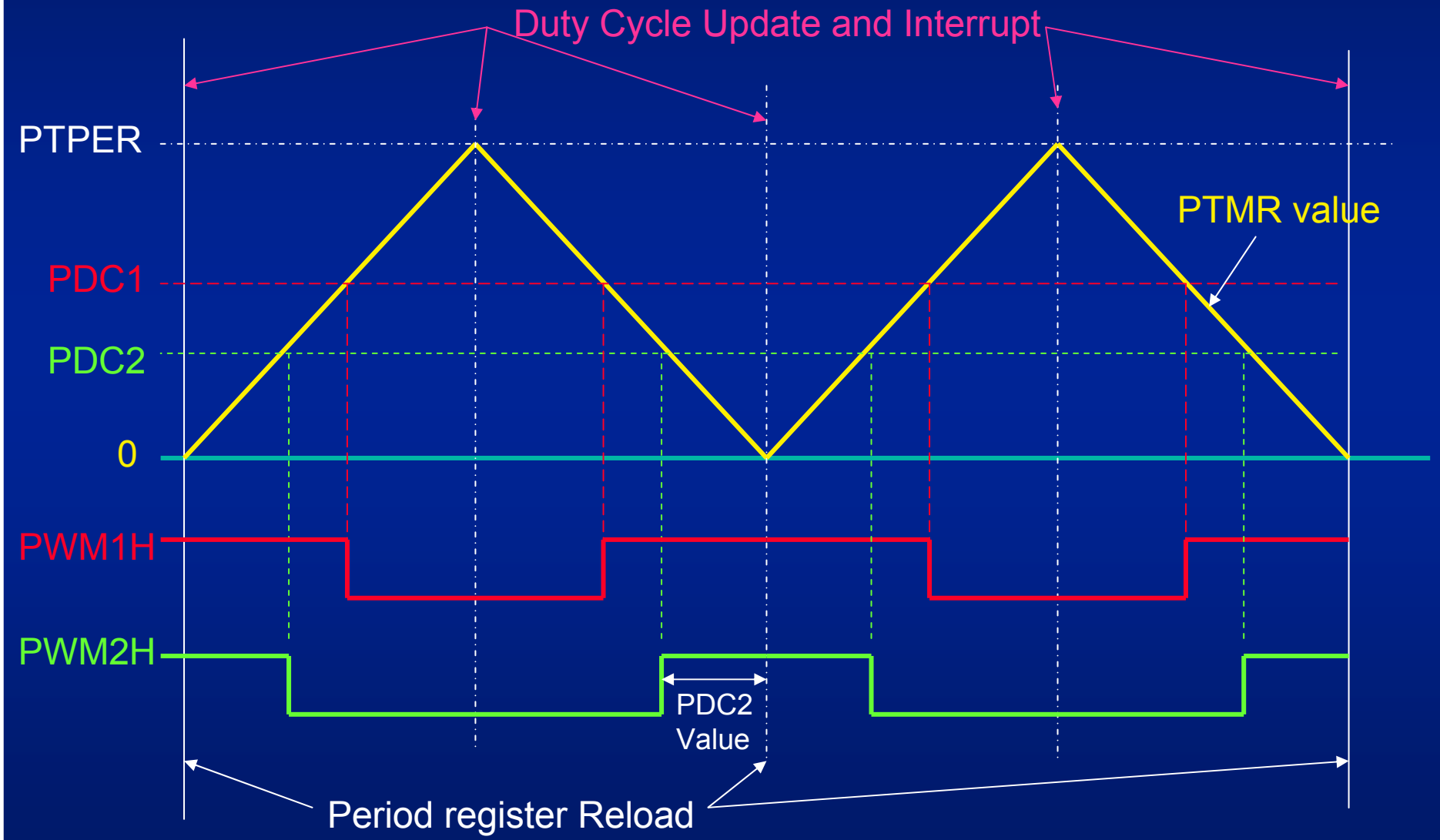
Single Event PWM



Center Aligned PWM



Center Aligned PWM with Double Update



PWM Interrupt

- Free Running
  - Interrupt – When the PTMR is reset to “0”
  - Period & Duty – PTMR is matched with PTPER
- Single Event
  - Interrupt – When the PTMR is reset to “0”
  - Period – PTMR is matched with PTPER
  - Duty – Set PTEN bit (PTMR=0)
- Up/Down Counting
  - Interrupt – When the PTMR count downwards to zero
  - Period and Duty - PTMR count downwards to zero
- Up/Down Counting with Double Updates
  - Interrupt – Each PTMR is equal zero and match with PTPER
  - Period – When the PTMR count downwards to zero
  - Duty – Each PTMR is equal zero and match with PTPER

MC PWM Period

- PWM Frequency and Counting Resolution
  - The 15-bit PTMR increments every Tcy
  - PTPER sets the counting period in Tcy
  - 16-bit duty cycles allow Tcy/2 edge resolution
  - Upper 15 bits of Duty Cycle compared to PTMR counter
  - LSbit of duty cycle specifies Tcy/2 boundary
  - Divide count period value by 2 to get PTPER
  - PWM frequency is halved for center-aligned

PWM Period Calculation

Edge Aligned : $PTPER = \frac{Fcy}{F_{PWM} * (PTMR\ Prescaler)} - 1$

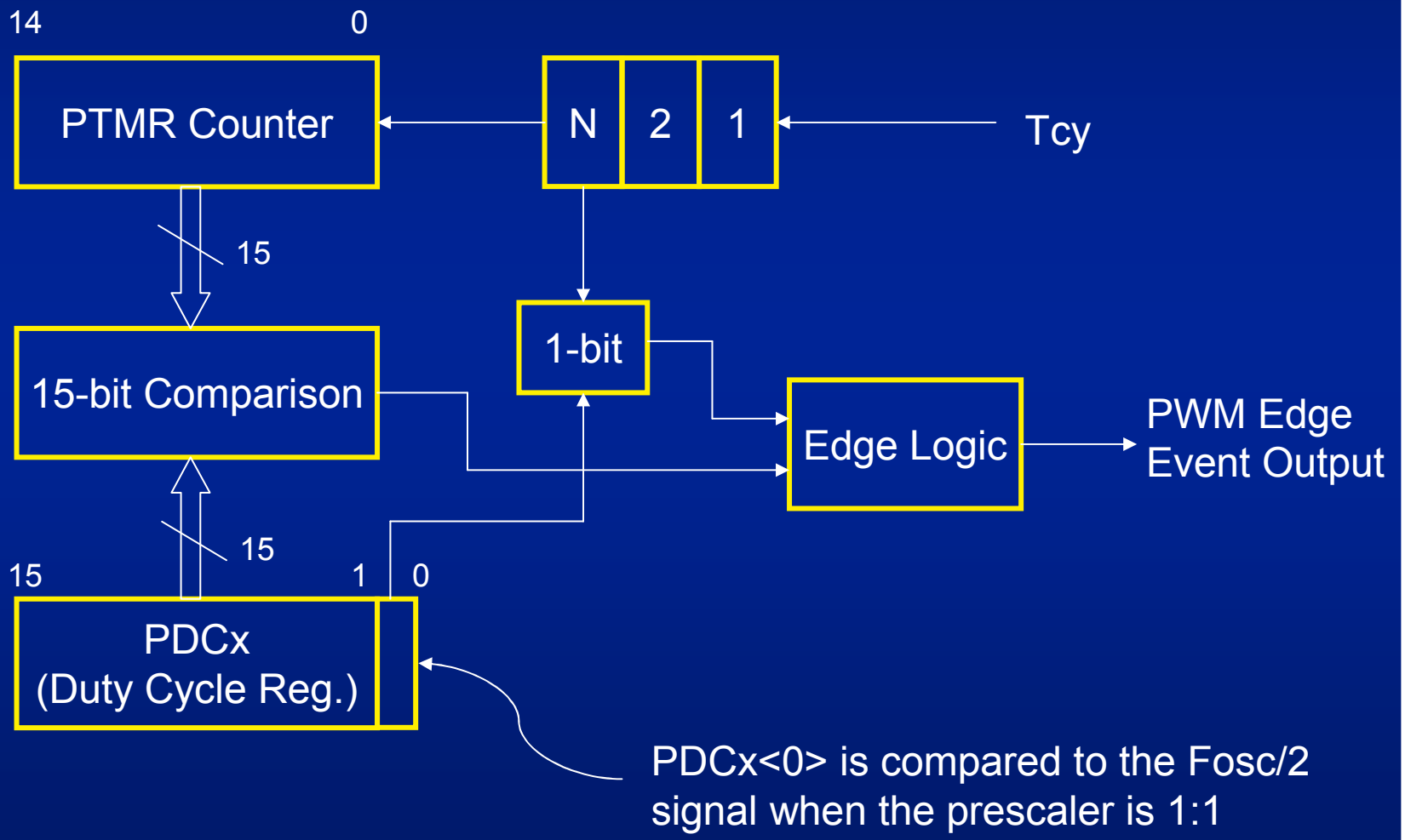
Center Aligned : $PTPER = \frac{\left[\frac{Fcy}{F_{PWM} * (PTMR\ Prescaler)} - 1\right]}{2}$

Edge Aligned Example :

Fosc = 80MHz , Fcy = 20MHz
F_{PWM} = 20.0KHz
PTMR Prescaler = 1:1

$$PTPER = \frac{20{,}000{,}000}{20{,}000 * 1} - 1 = 1000 - 1 = 999$$

Duty Cycle Comparison



Duty Cycle Resolution

- PWM Resolution for Center Aligned PWM

Resolution = log10(2*PTPER)/log10(2)

Tcy	PTPER Value	PWM Resolution	PWM Frequency
33nS(30MHz)	0x7FFF	16 bits	915Hz
33nS(30MHz)	0x3FF	11 bits	29.3KHz
50nS(20MHz)	0x7FFF	16 bits	610Hz
50nS(20MHz)	0x1FF	10 bits	39.1KHz
200nS(5MHz)	0x7FFF	16 bits	153Hz
200nS(5MHz)	0x7F	8 bits	39.1KHz

With 1:1 prescaler

PWM Control Register 1

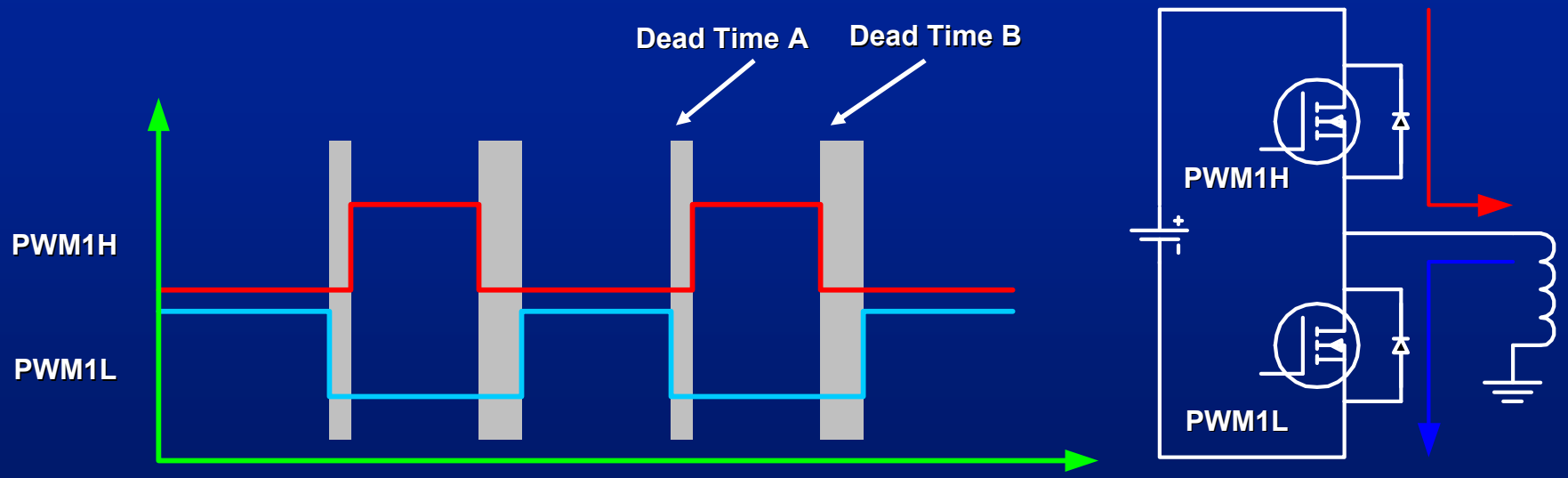
- PWMCON1 register
  - PMOD4:PMOD1 : PWM I/O Pair Mode selection bit
    - 1 = PWM I/O pin pair is in the independent Output
    - 0 = PWM I/O pin pair is in the complementary Output
  - PEN4H-PEN1H : PWMxH I/O Enable bit
    - 1 = PWMxH pin is enable for PWM output
    - 0 = PWMxH pin disabled, general purpose I/O
  - PEN4L-PEN1L : PWMxL I/O Enable bit
    - 1 = PWMxL pin is enable for PWM output
    - 0 = PWMxL pin disabled, general purpose I/O

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	-	-	**PMOD4**	**PMOD3**	**PMOD2**	**PMOD1**
bit15	14	13	12	11	10	9	bit8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PEN4H	**PEN3H**	**PEN2H**	**PEN1H**	**PEN4L**	**PEN3L**	**PEN2L**	**PEN1L**
bit7	6	5	4	3	2	1	bit0

MCPWM Dead Time Insertion

- Applies only to pin pairs in complimentary mode
- Two programmable dead times
- One dead time per pair for multiple inverters OR
- Two dead times per pair for distortion optimization
- T_{cy} minimum resolution with four pre-scale options



Dead Time Registers 1

- DTCON1 selects dead time value for A and B

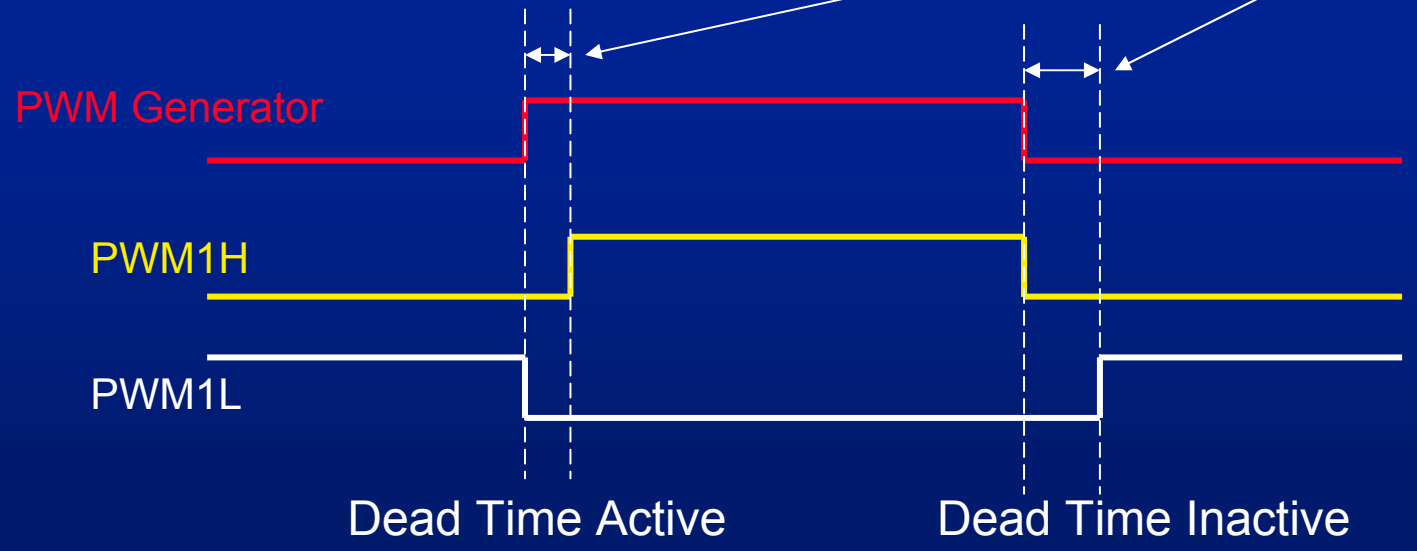
- DTBPS<1:0> : Dead Time Unit B Prescaler Select bits
  - 00 -> 1 Tcy , 01 -> 2 Tcy
  - 10 -> 4 Tcy , 11 -> 8 Tcy
- DBT<5:0> : Unsigned 6-bit Dead Time Value for Unit B
- DTAPS<1:0> : Dead Time Unit A Prescaler Select bits
- DAT<5:0> : Unsigned 6-bit Dead Time Value for Unit A

Dead Time Registers 2

- DTCON2 selects which dead time value is assigned to the active(A) or inactive(I) edge of each generator

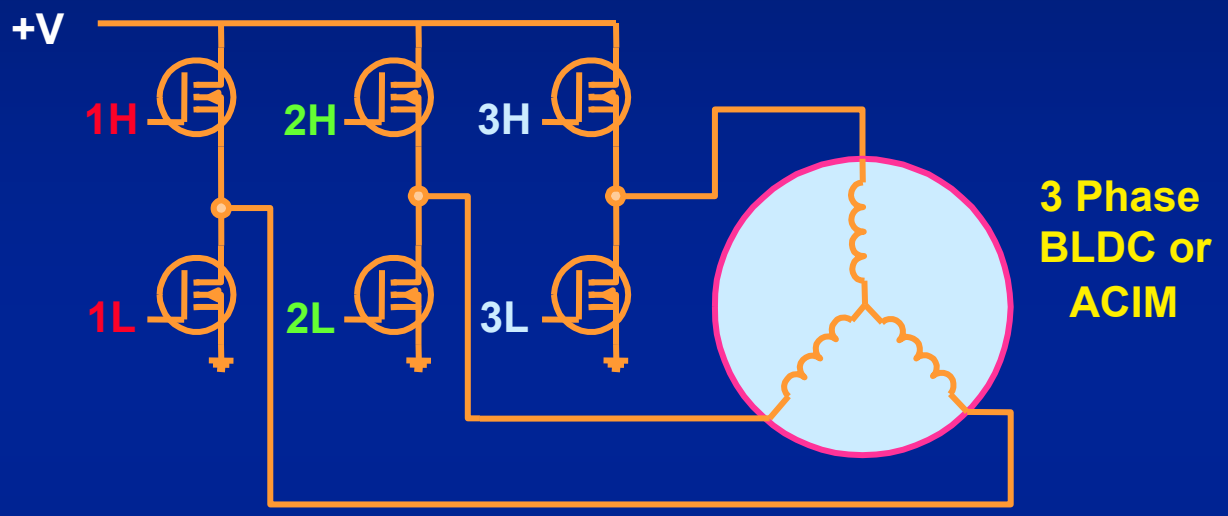
-	-	-	-	-	-	-	-
bit15	14	13	12	11	10	9	bit8

DTS4A	DTS4I	DTS3A	DTS3I	DTS2A	DTS2I	DTS1A	DTS1I
bit7	6	5	4	3	2	1	bit0

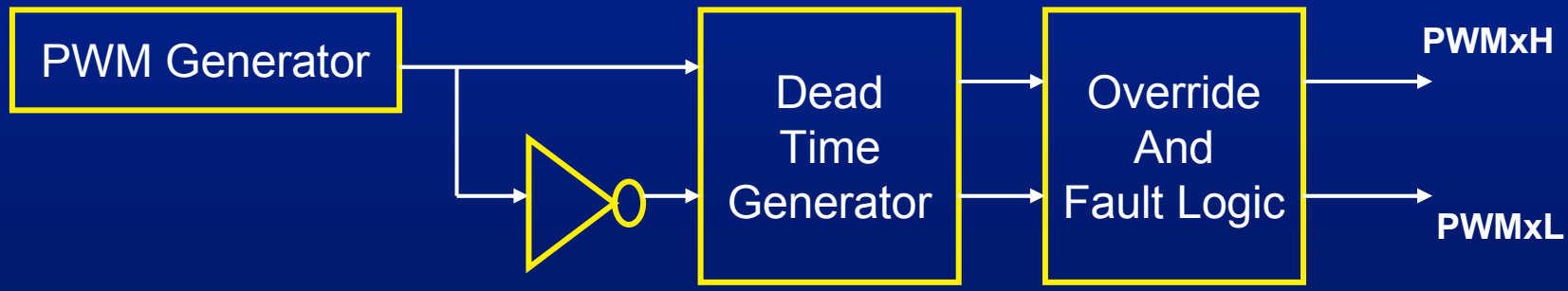


Complementary Output Mode

Typical Load for Complementary PWM Outputs

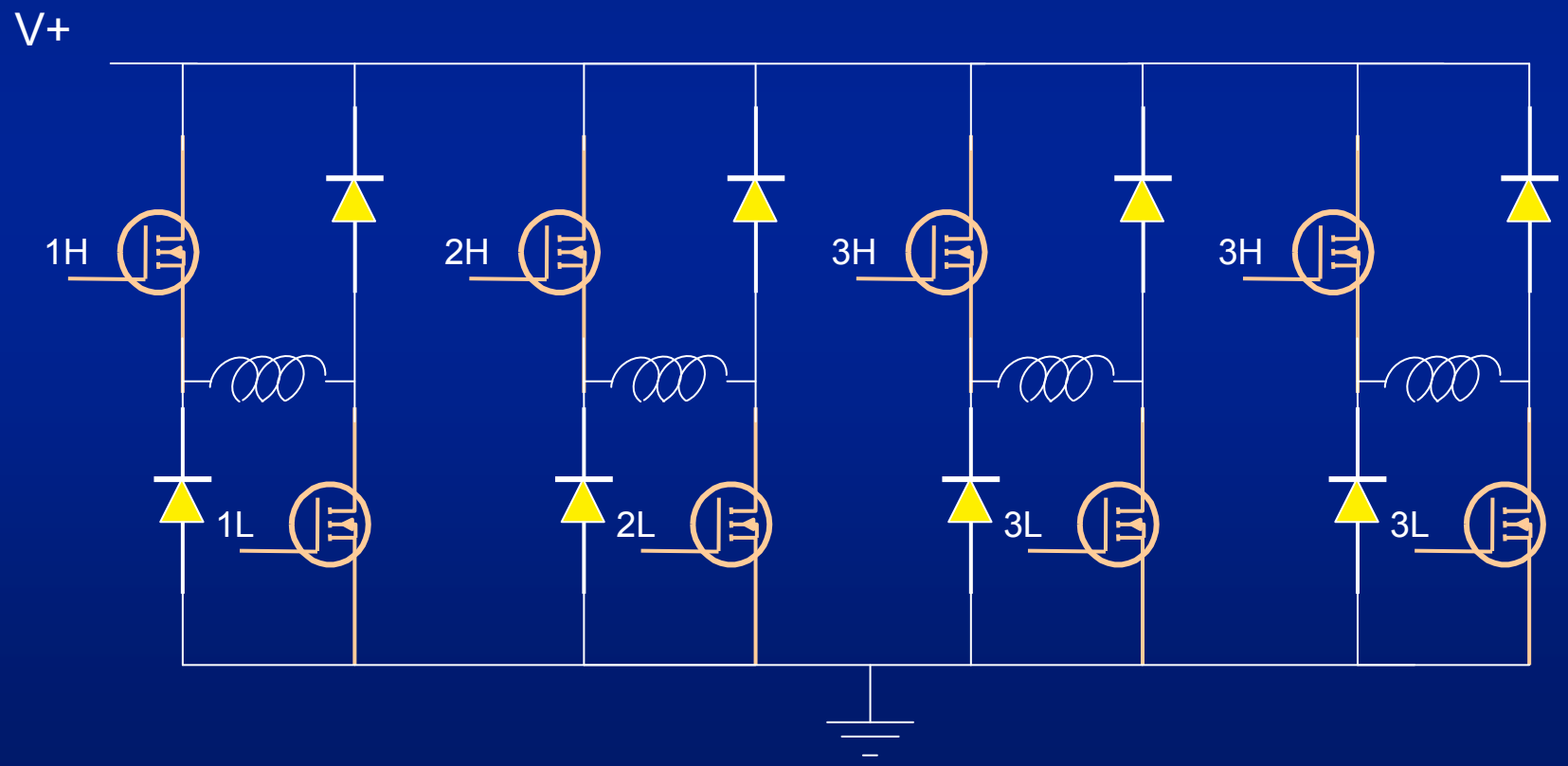


PWM Channel Block Diagram



Independent Outputs

- Independent mode is used for switched reluctance motors
- Special inverter circuit to control current in SR motors
- Independent mode enables both devices to turn on



Output Override

- Used for motor commutation
- Drive PWM or Drive active or inactive
- POVD bits decide if I/O pin is controlled manually
- POUT bits set state of I/O pin under manual control
- Dead time requirements are always satisfied incomplementary mode

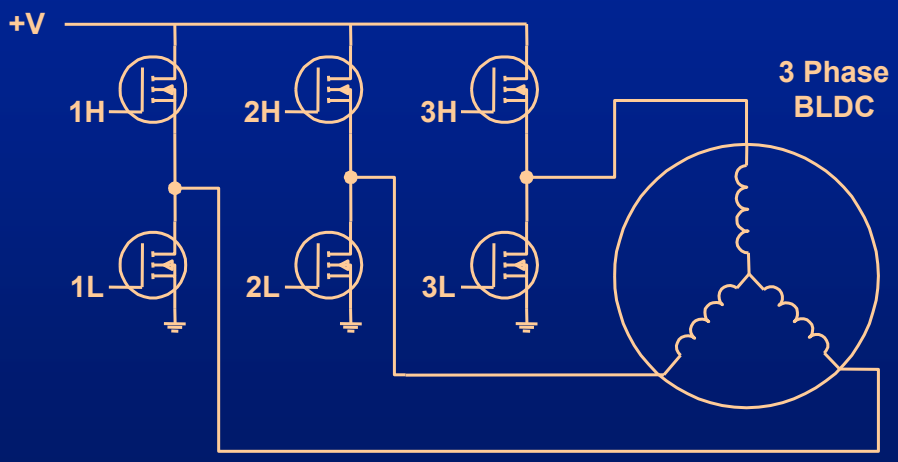
OVDCON Register

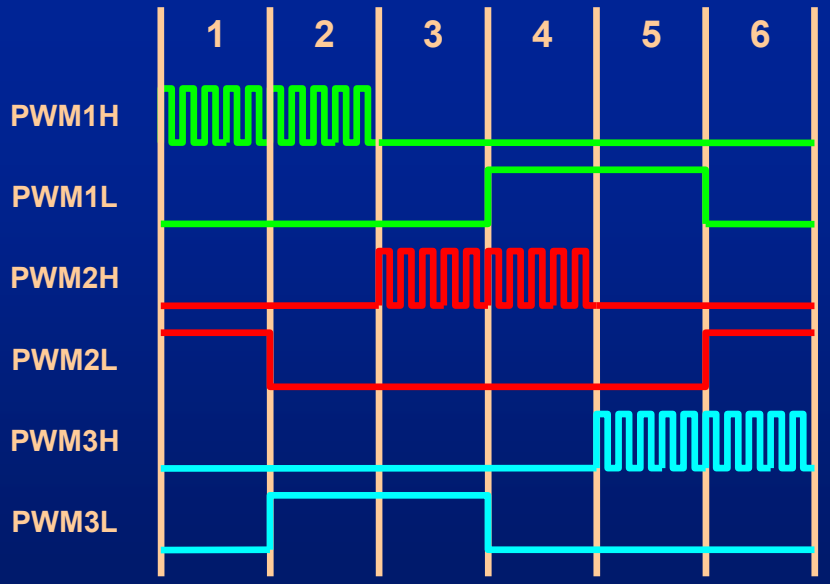
POVD4H	POVD4L	POVD3H	POVD3L	POVD2H	POVD2L	POVD1H	POVD1L
bit15	14	13	12	11	10	9	

POUT4H	POUT4L	POUT3H	POUT3L	POUT2H	POUT2L	POUT1H	POUT1L
bit7	6	5	4	3	2	1	

Output Override

- Using PWM for 6-step modulation
  - Energize BLDC motor coils based on measured rotor position
  - PWM override register used to control active transistors -- duty cycle sets current





Set the PWM output

- FBORPOR Register in the Configuration Words
  - Set the Motor Control PWM Module High Side Polarity
    - HPOL bit =1 , PWM Module high-side is active-high
    - HPOL bit =0 , PWM Module low-side is active-low
  - Set the Motor Control PWM Module Low Side Polarity
    - LPOL bit =1 , PWM Module high-side is active-high
    - LPOL bit =0 , PWM Module low-side is active-low

- OVDCON Register
  - POVD4H~POVD1L
    - =1 , PWMxx I/O pin is PWM output
    - =0 , PWMxx I/O pin is corresponding POUTxx bit
  - POUT4H~POUT1L
    - =1 , PWMxx I/O pin is driven ACTVE
    - =0 , PWMxx I/O pin is driven INACTIVE

MCPWM Fault Inputs

- Two programmable fault pins: Fault A, Fault B
- Fault pins can be assigned to each output pair
- Fault state for each output is programmable
- Automatic or latched fault protection
- Interrupt vector for each fault input
- Fault A has priority over Fault B
- Fault condition overrides all other pin control
- Fault pins can be used as interrupt pins when not used for PWM module

Motor Control PWM

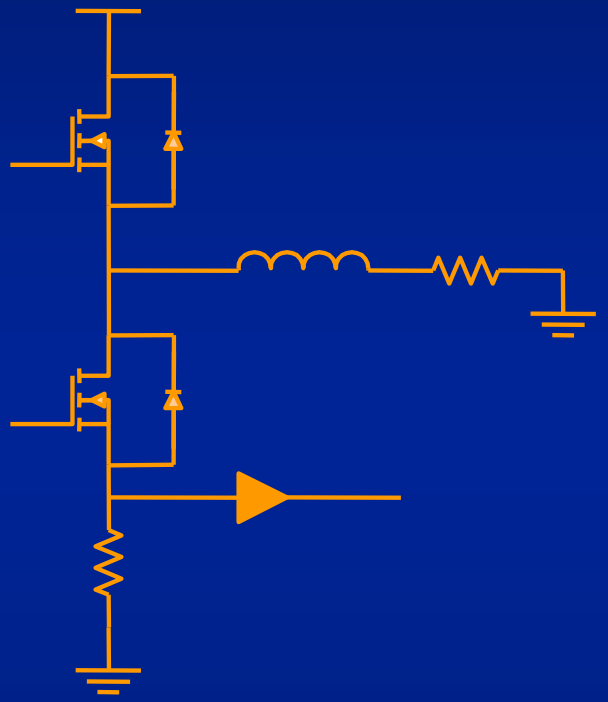
- FLTACON Register
  - FAENx - Enables pin pair for control by fault pin
  - FLTAM - Latched or cycle-by-cycle mode
  - FAOVxx - Override value bits
  - FLTBCON register is identical
  - For labs - FLTA drives all outputs inactive, latched mode

FAOV4H	FAOV4L	FAOV3H	FAOV3L	FAOV2H	FAOV2L	FAOV1H	FAOV1L
bit15	14	13	12	11	10	9	bit8

FLTAM	-	-	-	FAEN4	FAEN3	FAEN2	FAEN1
bit7	6	5	4	3	2	1	bit0

A/D Special Event Trigger

- SEVTCMP (special event compare) register sets A/D conversion start time
- SEVTDIR bit sets direction of PTMR
- Optional post-scaler



SEVTDIR	SEVTCMP<14:8>						
bit15	14	13	12	11	10	9	bit8

SEVTCMP<7:0>							
bit7	6	5	4	3	2	1	bit0

MC PWM in Power Save Modes

- In SLEEP mode all PWM output pins are frozen to the value prior to entering sleep
- Fault pins are active
- Fault event wakes the device from sleep
- In IDLE mode the PWM can optionally continue to operate
- Time base interupt wakes the devise from IDLE

MCPWM Lab 1

- Set the MCPWM as a PWM Complement Mode output with 10 bits resolution
- Prescaler 1:1 , Fosc = 7.3728MHz x 8 PLL
  - Period = 20KHz, PETR= [7372800 * 2 / (20000*1)] -1 = 736
- MCPWM counter mode : Edge Alignment Mode
- Use Dead Time A with : 16 Tcy
  - (dsPIC30F4011 has the Dead Time A only)
- VR1 connect with AN0 to measure the input voltage with 10 bits ADC resolution
- Used the ADC conversion result to present on the PWM output and show the Duty Cycle on LCD module

PWM Modulation

PWM Modulation

- We want to use PWM to drive AC motor
- AC motor needs a sine input at 60 Hz
- How do we get a 60 Hz sine wave from PWM with phase offsets?
- We will use lookup table method today

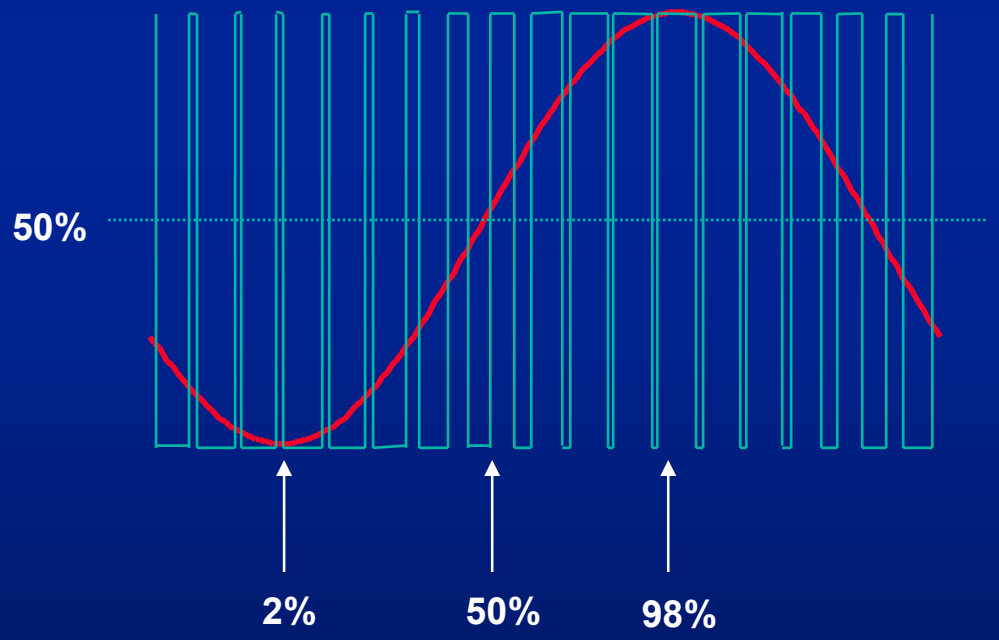
PWM Modulation

- Lookup Table
  - Use PWM module like a DAC
  - Create waveform table in memory
  - Periodically add value to table pointer to determine modulation frequency
  - Add constant offset to table pointer to get values for different motor phases
  - Retrieve signed value and scale for PWM
  - Signed lookup value added to 50% DC value

PWM Modulation

- Motor voltage is PWM at DC bus magnitude
- Motor Integrates PWM Voltage and produces Sinusoidal Current with small ripple at carrier frequency
- Lookup values scaled by PWM count period and modulation amplitude

Current Ripple = 4/(f*V*L)



PWM Modulation

- Setting the Modulation Frequency
  - 16-bit ‘Phase’ variable represents 360 degrees
  - Add value to Phase at PWM rate, 16 kHz
  - Delta value for 60 Hz

Hz/bit = (16,000/65536) = 0.244 Hz/bit

Delta_Phase = 60Hz/0.244 Hz/bit = 246 bits

Phase = Phase + Delta_Phase;

PWM Modulation

- Looking Up the Sine Values
  - 64 entry table contains 360 degree sine wave
  - Add offsets for phase differences
    - Binary arithmetic works well for 3-phase
  - Strip lower 10 bits of 'Phase' for lookup

```
Phase1_Index = Phase >> 10;

Phase2_Index = (Phase + 0x5555) >> 10;

Phase3_Index = (Phase + 0xAAAA) >> 10;
```

Sine Wave PWM Lab2

- Use the MC-PWM to generate the Sine Wave PWM, driven the AC Motor
- PWM r=Resolution : at least 10-bit
- PWM Carry Frequency : above 20 KHz
- Target AC Frequency : 60 Hz with 3 Phase
- Don’t concern the speed change

Hint Lab2

- Fcy = 7372800 * 2 (XT w/PLL 8x)
- PTPER = (Fcy/(FPWM*PTMR Prescaler)) – 1
  - PTPER = (14745600/(20000*1)) – 1 =736
  - Duty Cycle control Value : 0 to 1473 (737 * 2)
  - PWM Resolution above 10-bit (0 to 1023)
- Calculate the angle
  - 60 Hz = 1Sec / 60 Hz = 16666.6 us
  - Every 16666.6 us need to update duty from 0^0 to 360^0
  - 16666.6 * 7.3728 * 2 = 245760 (Request Tcy count)
  - Every Interrupt from Timer1, the output will increase 3^0
  - 245760 Tcy / ($360^0/3^0$) = 2048 Tcy (set for Timer1)
- Use the Scope to measure the 3 phase Sine-Wave PWM output that has 120^0 difference phase

PWM Amplitude

- Setting the Modulation Amplitude
  - 16-bit 'Amplitude' variable sets mod. amplitude
  - Also need to scale result to duty cycle range
    - Lookup value is signed

PDC = (sinetable[index] * Amplitude) / 32,768

PDC = (PDC * PTPER) / 32,768

PTPER = max_duty_cycle >> 1 = 50% duty cycle

Lab3 (Options)

- Modify the code from Lab 2
- Use the VR1 to adjust the Sine-Wave PWM output amplitude

Lab4 (Options)

- Modify the code from Lab3
- Add the VR2 to control the Sine-Wave PWM Frequency
- Hint, think about how to change the Timer1 using the ADC result from VR2