```
// This Pine Script® code is subject to the terms of the Mozilla Public License 2.0 at https://mozilla.org/MPL/2.0/

//@version=6

indicator("MFB - FVG-123 - Price Delta - CISD and Absorption", overlay = true, max_labels_count = 500)


// --- Constants ---

const color BULL_COLOR = #089981

const color BEAR_COLOR = #f23645

const color OTHER_COLOR = #5b9cf6


// --- Inputs ---

int footprintTicksInput = input.int(4, "Footprint ticks per row", minval = 1, maxval = 100, tooltip = "Price grouping used to read real-trade footprint delta on each chart bar.", group = "Order Flow")

int approachBarsInput = input.int(4, "Completed approach bars", minval = 3, maxval = 5, tooltip = "Completed bars immediately before the first FVG touch used for approach context.", group = "Order Flow")

float minDepartureDeltaFractionInput = input.float(0.10, "Minimum FVG confirmation delta / volume", minval = 0.01, maxval = 0.95, step = 0.01, tooltip = "FVG flow qualification threshold for the departure candle's directional delta as a fraction of its footprint volume.", group = "FVG-123 Qualification")

bool showFvgLabelsInput = input.bool(true, "Show FVG-123 labels", tooltip = "Show labels when an FVG touch-and-reclaim sequence completes.", group = "Display")
```

bool enableAbsorptionInput = input.bool(true, "Enable potential absorption signals", tooltip = "Detect potential absorption from elevated footprint activity, directional delta, and a rejecting candle shape.", group = "Absorption")

int absorptionLookbackInput = input.int(20, "Volume average lookback", minval = 5, maxval = 200, tooltip = "Bars used to calculate baseline footprint volume.", group = "Absorption")

float absorptionVolumeMultipleInput = input.float(1.5, "Minimum volume vs average", minval = 1.0, maxval = 10.0, step = 0.1, tooltip = "Signal requires footprint volume at least this multiple of its recent average.", group = "Absorption")

float absorptionDeltaFractionInput = input.float(0.20, "Minimum opposing delta / volume", minval = 0.01, maxval = 0.95, step = 0.01, tooltip = "Minimum absolute delta fraction showing aggressive flow into the potential absorbing side.", group = "Absorption")

float absorptionWickFractionInput = input.float(0.35, "Minimum rejection wick / range", minval = 0.05, maxval = 0.95, step = 0.05, tooltip = "Require a lower wick for bullish absorption or an upper wick for bearish absorption, as a fraction of the candle range.", group = "Absorption")

int absorptionAtrLengthInput = input.int(14, "ATR length", minval = 2, maxval = 200, tooltip = "ATR lookback used to limit signals to bars without an unusually large price range.", group = "Absorption")

float absorptionMaxRangeAtrInput = input.float(1.0, "Maximum bar range (ATR)", minval = 0.1, maxval = 5.0, step = 0.1, tooltip = "The candle range must be no larger than this multiple of ATR to count as limited price progress.", group = "Absorption")

bool enableCisdInput = input.bool(true, "Enable CISD price/delta signals", tooltip = "Detect a close through the most recent opposing candle's open, confirmed by directional footprint delta.", group = "CISD")

int cisdReferenceMaxAgeInput = input.int(8, "Maximum reference age (bars)", minval = 1, maxval = 500, tooltip = "Ignore the opposing-candle open if it is older than this number of chart bars.", group = "CISD")

float cisdDeltaFractionInput = input.float(0.20, "Minimum CISD delta / volume", minval = 0.01, maxval = 0.95, step = 0.01, tooltip = "Minimum directional footprint delta as a fraction of volume on the CISD break candle.", group = "CISD")

int cisdSwingLookbackInput = input.int(5, "Swing break lookback", minval = 2, maxval = 50, tooltip = "Require the CISD close to break the high or low of this many prior bars.", group = "CISD")

int cisdAtrLengthInput = input.int(14, "Displacement ATR length", minval = 2, maxval = 200, tooltip = "ATR lookback used to evaluate the CISD candle's directional body size.", group = "CISD")

float cisdMinBodyAtrInput = input.float(0.35, "Minimum body size (ATR)", minval = 0.05, maxval = 3.0, step = 0.05, tooltip = "Require the directional CISD candle body to meet this multiple of ATR.", group = "CISD")

bool showCisdLabelsInput = input.bool(true, "Show CISD labels", tooltip = "Show CISD level, delta, and delta/volume on qualified events.", group = "Display")

bool showAbsorptionLabelsInput = input.bool(true, "Show absorption labels", tooltip = "Show potential bullish and bearish absorption labels.", group = "Display")

color bullColorInput = input.color(BULL_COLOR, "Bullish", tooltip = "Color for bullish events.", group = "Style")

color bearColorInput = input.color(BEAR_COLOR, "Bearish", tooltip = "Color for bearish events.", group = "Style")

color otherColorInput = input.color(OTHER_COLOR, "Not qualified / unavailable", tooltip = "Color for events that fail flow qualification or lack required footprint data.", group = "Style")

```
string labelSizeInput = input.string("Normal", "Label text size", options =
["Tiny", "Small", "Normal", "Large", "Huge"], tooltip = "Text size used by event
labels.", group = "Style")


// --- Formatting helpers ---
f_num(float value) =>
    na(value) ? "n/a" : str.tostring(value, format.volume)


f_pct(float value) =>
    na(value) ? "n/a" : str.tostring(value * 100.0, "#.0") + "%"


f_label_size(string sizeName) =>
    switch sizeName
        "Tiny" => size.tiny
        "Small" => size.small
        "Normal" => size.normal
        "Large" => size.large
        => size.huge


labelSize = f_label_size(labelSizeInput)


// --- Footprint measurements ---
footprint footprintData = request.footprint(footprintTicksInput)
float barDelta = na(footprintData) ? na : footprint.delta(footprintData)
```

```
float barVolume = na(footprintData) ? na :
footprint.total_volume(footprintData)

bool barFlowValid = not na(barDelta) and not na(barVolume) and barVolume >
0.0

float departureDeltaFraction = barFlowValid ? barDelta / barVolume : na

float approachCoverage = ta.sma(barFlowValid ? 1.0 : 0.0,
approachBarsInput)[1]

float approachDeltaSum = ta.sma(nz(barDelta, 0.0), approachBarsInput)[1] *
approachBarsInput

float approachVolumeSum = ta.sma(nz(barVolume, 0.0),
approachBarsInput)[1] * approachBarsInput

float approachDeltaFraction = not na(approachVolumeSum) and
approachVolumeSum > 0.0 ? approachDeltaSum / approachVolumeSum : na

float approachTravel = ta.sma(math.abs(close - close[1]),
approachBarsInput)[1] * approachBarsInput

float approachNetMove = close[1] - close[approachBarsInput + 1]

float approachEfficiency = not na(approachTravel) and approachTravel > 0.0 ?
math.abs(approachNetMove) / approachTravel : na

bool approachDataValid = not na(approachCoverage) and approachCoverage
>= 0.999 and not na(approachVolumeSum) and approachVolumeSum > 0.0


// --- Latest active bullish and bearish FVG-123 candidates ---

var bool bullActive = false

var float bullTop = na

var float bullBottom = na

var int bullCreatedBar = na
```

```
var bool bullTouched = false
var int bullTouchBar = na
var float bullInteractionDelta = 0.0
var float bullInteractionVolume = 0.0
var bool bullInteractionValid = false
var float bullMaxPenetration = 0.0
var float bullApproachDelta = na
var float bullApproachFraction = na
var float bullApproachEfficiency = na
var bool bullApproachValid = false

var bool bearActive = false
var float bearTop = na
var float bearBottom = na
var int bearCreatedBar = na
var bool bearTouched = false
var int bearTouchBar = na
var float bearInteractionDelta = 0.0
var float bearInteractionVolume = 0.0
var bool bearInteractionValid = false
var float bearMaxPenetration = 0.0
var float bearApproachDelta = na
var float bearApproachFraction = na
```

```
var float bearApproachEfficiency = na
var bool bearApproachValid = false

bool bullishFvg123Signal = false
bool bearishFvg123Signal = false

if barstate.isconfirmed and bullActive
   if low <= bullBottom
      bullActive := false
   else if bullTouched and bar_index > bullTouchBar and close > bullTop
      bool bullStoryDataValid = bullApproachValid and bullInteractionValid and barFlowValid
      bool bullFlowPass = bullStoryDataValid and departureDeltaFraction >= minDepartureDeltaFractionInput
      bullishFvg123Signal := bullFlowPass
      string bullText = bullFlowPass ? "Bullish FVG-123\nDelta/Vol " + f_pct(departureDeltaFraction) : not bullStoryDataValid ? "Bullish FVG-123\nFootprint unavailable" : "Bullish FVG-123\nFlow not qualified\nDelta/Vol " + f_pct(departureDeltaFraction)
      color bullLabelColor = bullFlowPass ? bullColorInput : otherColorInput
      if showFvgLabelsInput
         label.new(x = math.max(0, bullCreatedBar - 2), y = math.avg(bullTop, bullBottom), text = bullText, xloc = xloc.bar_index, yloc = yloc.price, color = color.new(bullLabelColor, 10), style = label.style_label_right, textcolor = color.white, size = labelSize)
      bullActive := false
```

```
    else if not bullTouched and bar_index > bullCreatedBar and high >=
bullBottom and low <= bullTop
        bullTouched := true
        bullTouchBar := bar_index
        bullInteractionDelta := nz(barDelta, 0.0)
        bullInteractionVolume := nz(barVolume, 0.0)
        bullInteractionValid := barFlowValid
        bullMaxPenetration := math.min(math.max(bullTop - low, 0.0), bullTop -
bullBottom)
        bullApproachDelta := approachDeltaSum
        bullApproachFraction := approachDeltaFraction
        bullApproachEfficiency := approachEfficiency
        bullApproachValid := approachDataValid
    else if bullTouched
        bullInteractionDelta += nz(barDelta, 0.0)
        bullInteractionVolume += nz(barVolume, 0.0)
        bullInteractionValid := bullInteractionValid and barFlowValid
        bullMaxPenetration := math.max(bullMaxPenetration,
math.min(math.max(bullTop - low, 0.0), bullTop - bullBottom))


if barstate.isconfirmed and bearActive
    if high >= bearTop
        bearActive := false
    else if bearTouched and bar_index > bearTouchBar and close < bearBottom
```

```
    bool bearStoryDataValid = bearApproachValid and bearInteractionValid and barFlowValid

    bool bearFlowPass = bearStoryDataValid and departureDeltaFraction <= -minDepartureDeltaFractionInput

    bearishFvg123Signal := bearFlowPass

    string bearText = bearFlowPass ? "Bearish FVG-123\nDelta/Vol " + f_pct(departureDeltaFraction) : not bearStoryDataValid ? "Bearish FVG-123\nFootprint unavailable" : "Bearish FVG-123\nFlow not qualified\nDelta/Vol " + f_pct(departureDeltaFraction)

    color bearLabelColor = bearFlowPass ? bearColorInput : otherColorInput

    if showFvgLabelsInput

      label.new(x = math.max(0, bearCreatedBar - 2), y = math.avg(bearTop, bearBottom), text = bearText, xloc = xloc.bar_index, yloc = yloc.price, color = color.new(bearLabelColor, 10), style = label.style_label_right, textcolor = color.white, size = labelSize)

    bearActive := false

  else if not bearTouched and bar_index > bearCreatedBar and high >= bearBottom and low <= bearTop

    bearTouched := true

    bearTouchBar := bar_index

    bearInteractionDelta := nz(barDelta, 0.0)

    bearInteractionVolume := nz(barVolume, 0.0)

    bearInteractionValid := barFlowValid

    bearMaxPenetration := math.min(math.max(high - bearBottom, 0.0), bearTop - bearBottom)

    bearApproachDelta := approachDeltaSum
```

```
        bearApproachFraction := approachDeltaFraction
        bearApproachEfficiency := approachEfficiency
        bearApproachValid := approachDataValid
    else if bearTouched
        bearInteractionDelta += nz(barDelta, 0.0)
        bearInteractionVolume += nz(barVolume, 0.0)
        bearInteractionValid := bearInteractionValid and barFlowValid
        bearMaxPenetration := math.max(bearMaxPenetration,
math.min(math.max(high - bearBottom, 0.0), bearTop - bearBottom))


if barstate.isconfirmed and bar_index >= 2 and low > high[2]
    bullTop := low
    bullBottom := high[2]
    bullCreatedBar := bar_index
    bullTouched := false
    bullTouchBar := na
    bullInteractionDelta := 0.0
    bullInteractionVolume := 0.0
    bullInteractionValid := false
    bullMaxPenetration := 0.0
    bullApproachDelta := na
    bullApproachFraction := na
    bullApproachEfficiency := na
```

```
    bullApproachValid := false
    bullActive := true

if barstate.isconfirmed and bar_index >= 2 and high < low[2]
    bearTop := low[2]
    bearBottom := high
    bearCreatedBar := bar_index
    bearTouched := false
    bearTouchBar := na
    bearInteractionDelta := 0.0
    bearInteractionVolume := 0.0
    bearInteractionValid := false
    bearMaxPenetration := 0.0
    bearApproachDelta := na
    bearApproachFraction := na
    bearApproachEfficiency := na
    bearApproachValid := false
    bearActive := true

// --- Potential absorption detection ---
float averageFootprintVolume = ta.sma(barFlowValid ? barVolume : na,
absorptionLookbackInput)
float absorptionAtr = ta.atr(absorptionAtrLengthInput)
```

```
float barRange = high - low

float lowerWick = math.min(open, close) - low

float upperWick = high - math.max(open, close)

bool absorptionContextValid = barFlowValid and not
na(averageFootprintVolume) and averageFootprintVolume > 0.0 and not
na(absorptionAtr) and barRange > 0.0

bool bullishAbsorptionSignal = barstate.isconfirmed and
enableAbsorptionInput and absorptionContextValid and barVolume >=
averageFootprintVolume * absorptionVolumeMultipleInput and
departureDeltaFraction <= -absorptionDeltaFractionInput and lowerWick /
barRange >= absorptionWickFractionInput and close >= low + barRange * 0.5
and barRange <= absorptionAtr * absorptionMaxRangeAtrInput

bool bearishAbsorptionSignal = barstate.isconfirmed and
enableAbsorptionInput and absorptionContextValid and barVolume >=
averageFootprintVolume * absorptionVolumeMultipleInput and
departureDeltaFraction >= absorptionDeltaFractionInput and upperWick /
barRange >= absorptionWickFractionInput and close <= low + barRange * 0.5
and barRange <= absorptionAtr * absorptionMaxRangeAtrInput


// --- CISD price/delta detection ---

var float lastBearishCandleOpen = na

var int lastBearishCandleBar = na

var float lastBullishCandleOpen = na

var int lastBullishCandleBar = na

float bullishCisdLevel = lastBearishCandleOpen

float bearishCisdLevel = lastBullishCandleOpen
```

bool bullishCisdReferenceValid = not na(lastBearishCandleBar) and bar_index - lastBearishCandleBar <= cisdReferenceMaxAgeInput

bool bearishCisdReferenceValid = not na(lastBullishCandleBar) and bar_index - lastBullishCandleBar <= cisdReferenceMaxAgeInput

bool bullishCisdPriceBreak = bullishCisdReferenceValid and close > bullishCisdLevel and close[1] <= bullishCisdLevel

bool bearishCisdPriceBreak = bearishCisdReferenceValid and close < bearishCisdLevel and close[1] >= bearishCisdLevel

float cisdAtr = ta.atr(cisdAtrLengthInput)

float cisdPriorSwingHigh = ta.highest(high[1], cisdSwingLookbackInput)

float cisdPriorSwingLow = ta.lowest(low[1], cisdSwingLookbackInput)

bool bullishCisdSwingBreak = not na(cisdPriorSwingHigh) and close > cisdPriorSwingHigh

bool bearishCisdSwingBreak = not na(cisdPriorSwingLow) and close < cisdPriorSwingLow

bool bullishCisdDisplacement = not na(cisdAtr) and cisdAtr > 0.0 and close > open and close - open >= cisdAtr * cisdMinBodyAtrInput

bool bearishCisdDisplacement = not na(cisdAtr) and cisdAtr > 0.0 and close < open and open - close >= cisdAtr * cisdMinBodyAtrInput

bool bullishCisdSignal = barstate.isconfirmed and enableCisdInput and bullishCisdPriceBreak and bullishCisdSwingBreak and bullishCisdDisplacement and barFlowValid and departureDeltaFraction >= cisdDeltaFractionInput

bool bearishCisdSignal = barstate.isconfirmed and enableCisdInput and bearishCisdPriceBreak and bearishCisdSwingBreak and bearishCisdDisplacement and barFlowValid and departureDeltaFraction <= -cisdDeltaFractionInput

```
if barstate.isconfirmed and close < open
   lastBearishCandleOpen := open
   lastBearishCandleBar := bar_index
if barstate.isconfirmed and close > open
   lastBullishCandleOpen := open
   lastBullishCandleBar := bar_index

if showAbsorptionLabelsInput and bullishAbsorptionSignal
   string bullAbsorptionText = "Potential bull absorption\nDelta " +
f_num(barDelta) + " (" + f_pct(departureDeltaFraction) + ")\nVolume " +
f_num(barVolume)
   label.new(bar_index, low, bullAbsorptionText, xloc = xloc.bar_index, yloc =
yloc.price, color = color.new(bullColorInput, 10), style = label.style_label_up,
textcolor = color.white, size = size.small)
if showAbsorptionLabelsInput and bearishAbsorptionSignal
   string bearAbsorptionText = "Potential bear absorption\nDelta " +
f_num(barDelta) + " (" + f_pct(departureDeltaFraction) + ")\nVolume " +
f_num(barVolume)
   label.new(bar_index, high, bearAbsorptionText, xloc = xloc.bar_index, yloc =
yloc.price, color = color.new(bearColorInput, 10), style =
label.style_label_down, textcolor = color.white, size = size.small)

if showCisdLabelsInput and bullishCisdSignal
   string bullCisdText = "Bullish CISD\nLevel " + str.tostring(bullishCisdLevel,
format.mintick) + "\nDelta " + f_num(barDelta) + " (" +
f_pct(departureDeltaFraction) + ")"
```

```
    label.new(bar_index, low, bullCisdText, xloc = xloc.bar_index, yloc =
yloc.price, color = color.new(bullColorInput, 5), style = label.style_label_up,
textcolor = color.white, size = size.small)

if showCisdLabelsInput and bearishCisdSignal

    string bearCisdText = "Bearish CISD\nLevel " + str.tostring(bearishCisdLevel,
format.mintick) + "\nDelta " + f_num(barDelta) + " (" +
f_pct(departureDeltaFraction) + ")"

    label.new(bar_index, high, bearCisdText, xloc = xloc.bar_index, yloc =
yloc.price, color = color.new(bearColorInput, 5), style =
label.style_label_down, textcolor = color.white, size = size.small)


// --- Alerts ---

alertcondition(bullishFvg123Signal, title = "Bullish FVG-123 delta", message =
"Bullish FVG-123 price/delta confirmation on {{ticker}} ({{interval}}).")

alertcondition(bearishFvg123Signal, title = "Bearish FVG-123 delta", message
= "Bearish FVG-123 price/delta confirmation on {{ticker}} ({{interval}}).")

alertcondition(bullishFvg123Signal or bearishFvg123Signal, title = "Any FVG-
123 delta", message = "FVG-123 price/delta confirmation on {{ticker}}
({{interval}}).")

alertcondition(bullishAbsorptionSignal, title = "Potential bullish absorption",
message = "Potential bullish footprint absorption on {{ticker}} ({{interval}}).")

alertcondition(bearishAbsorptionSignal, title = "Potential bearish absorption",
message = "Potential bearish footprint absorption on {{ticker}} ({{interval}}).")

alertcondition(bullishAbsorptionSignal or bearishAbsorptionSignal, title =
"Potential absorption", message = "Potential footprint absorption on {{ticker}}
({{interval}}).")

alertcondition(bullishCisdSignal, title = "Bullish CISD price/delta", message =
"Bullish CISD price/delta confirmation on {{ticker}} ({{interval}}).")
```

```
alertcondition(bearishCisdSignal, title = "Bearish CISD price/delta", message = "Bearish CISD price/delta confirmation on {{ticker}} ({{interval}}).")

alertcondition(bullishCisdSignal or bearishCisdSignal, title = "Any CISD price/delta", message = "CISD price/delta confirmation on {{ticker}} ({{interval}}).")
```