SANDRIDGE
THE POWER OF US™

Triple Combo/MRIL
Kansas Petrophysical Analysis

API WELL NO. 15033217020000

CLIENT	
COMPANY	SANDRIDGE EXPLORATION AND PRODUCTION LLC
WELL	ARIANA 3419 1-7H
FIELD	Mississippi Lime
COUNTY	COMANCHE
STATE	KANSAS

LOCATION : TWP: 34 S - Range: 19 W - Sec. 6
210 FSL 685 FWL
LAT
BHL

OTHER COMPUTATIONS
Rw = 0.035

PERMANENT DATUM GR ELEV. -999.25
LOG MEASURED FROM KB , ABOVE PERM DATUM
DRILLING MEASURED FROM KB

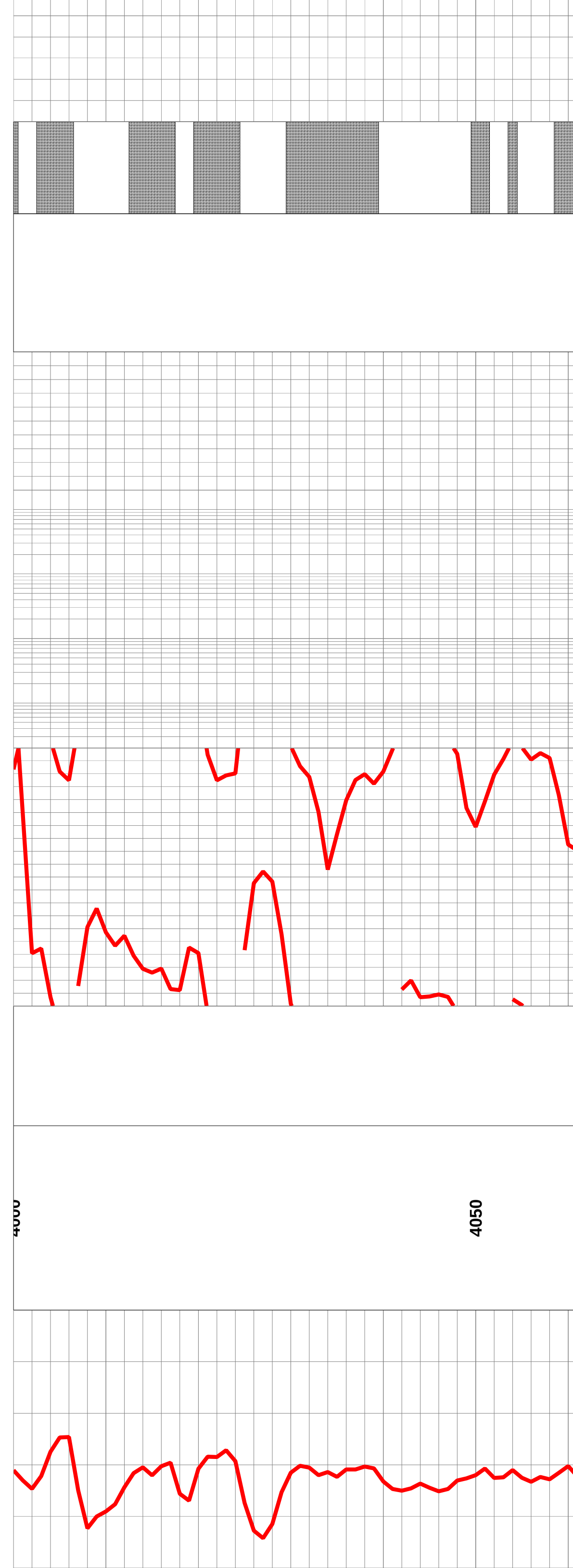
ELEV K.B. 1829
D.F.
G.L. 1809

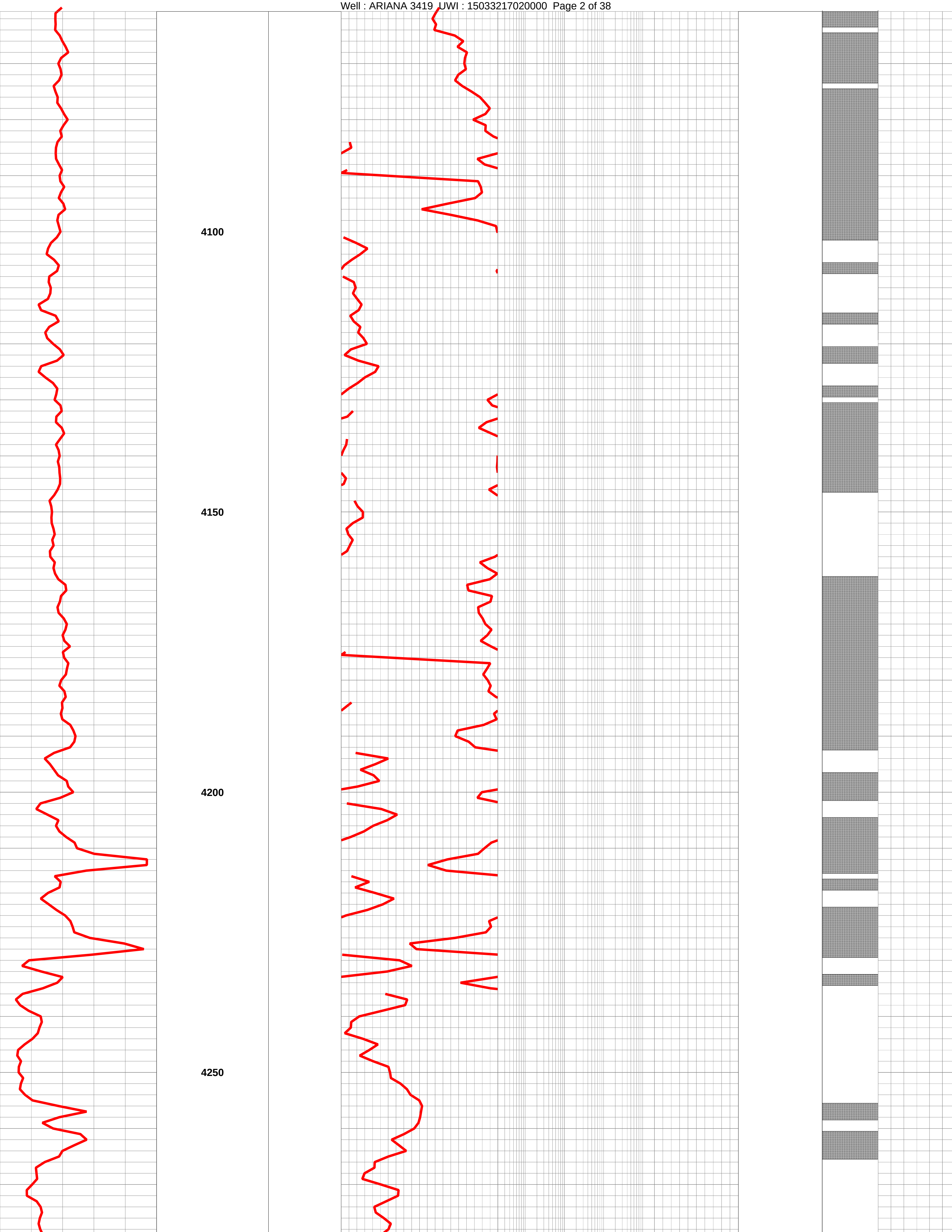
DATE	12 - APR - 2013			
RUN NO.				
DEPTH - DRILLER				
DEPTH - LOGGER	5737			
BOTTOM LOGGED INTERVAL	5737			
TOP LOGGED INTERVAL	3339			
CASING - DRILLER				
CASING - LOGGER				
BIT SIZE				
TYPE FLUID IN HOLE				
CASING SIZE / CASING WEIGHT				
DENSITY / VISCOSITY				
PH / FLUID LOSS				
SOURCE OF SAMPLE				
R_M @ MEASURED TEMP.	@	@	@	@
R_{MF} @ MEASURED TEMP.	@	@	@	@
R_{MC} @ MEASURED TEMP.	@	@	@	@
SOURCE : R_{MF} R_{MC}				
R_M @ BHT	@	@	@	@
CIRCULATION STOP DATE				
MAX. REC. TEMP				
Company / LOCATION	Archer			
RECORDED BY				
WITNESSED BY				

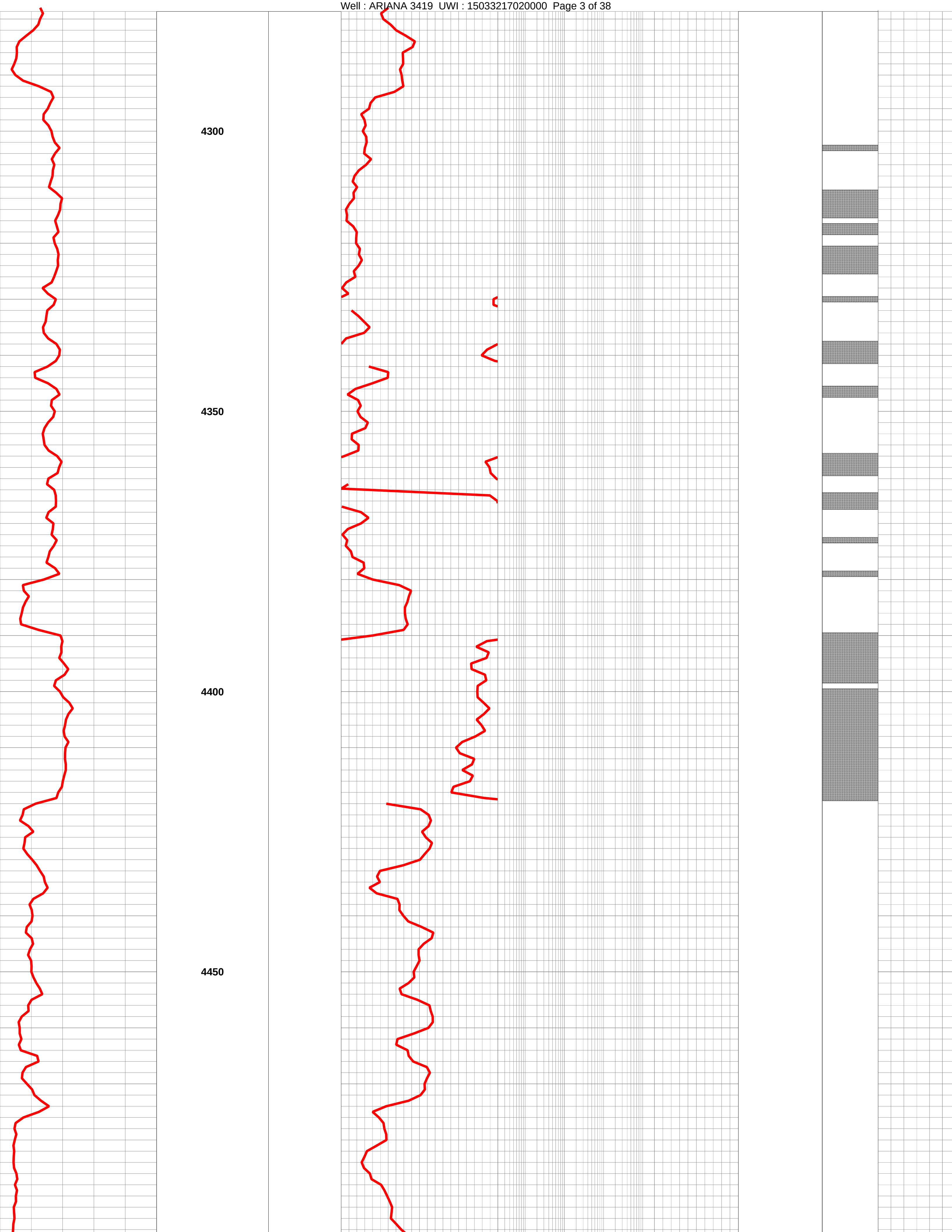
FOLD HERE

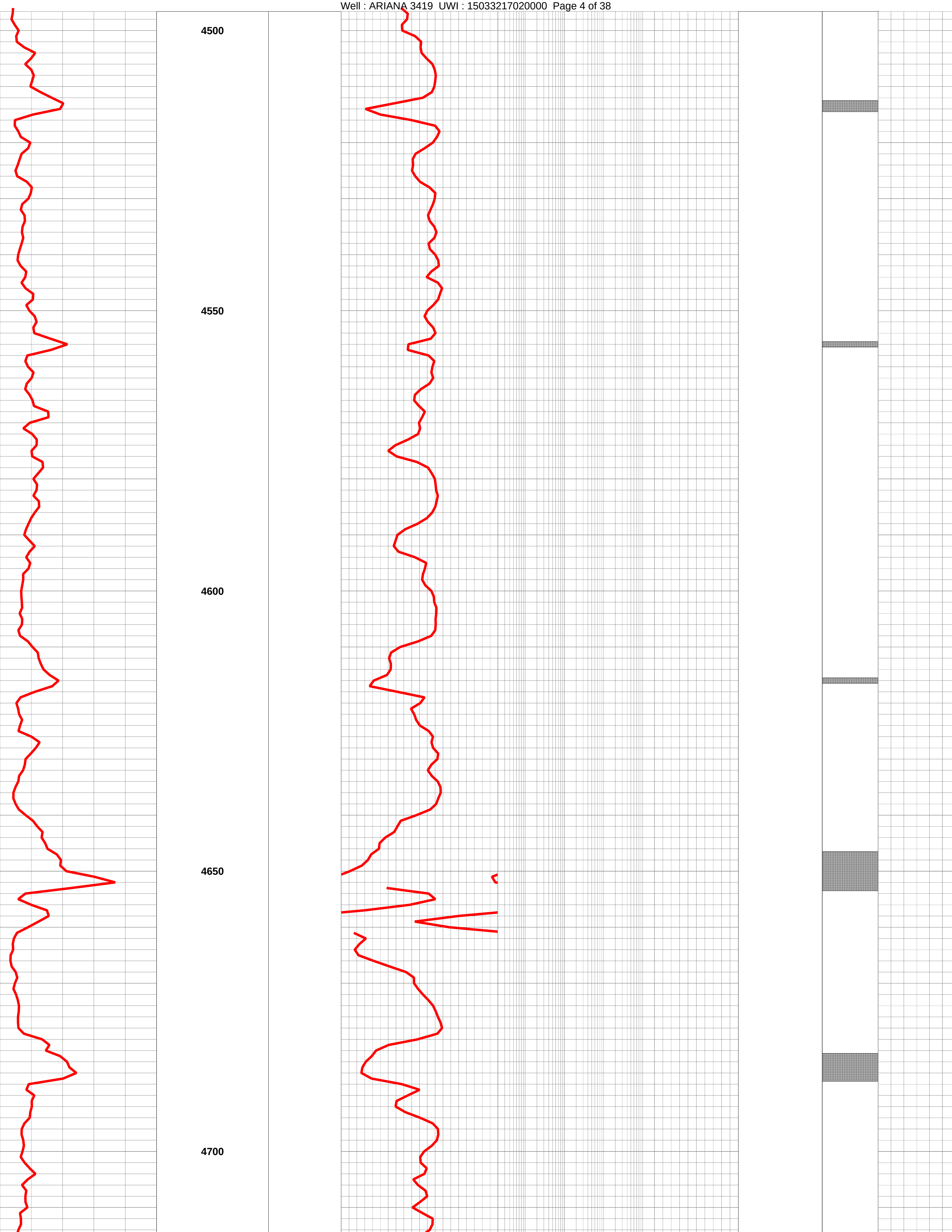
Interpretations are opinions based upon inferences from electrical or other measurements and algorithms, empirical relationships, and assumptions which are not infallible and with respect to which log analysts may differ. Accordingly, SandRidge Energy, Inc cannot and does not guarantee the accuracy or correctness of any interpretation and shall not be liable or responsible for any losses, costs, damages, or expenses incurred or sustained by anyone resulting from any interpretation made by any of our officers, agents, or employees.

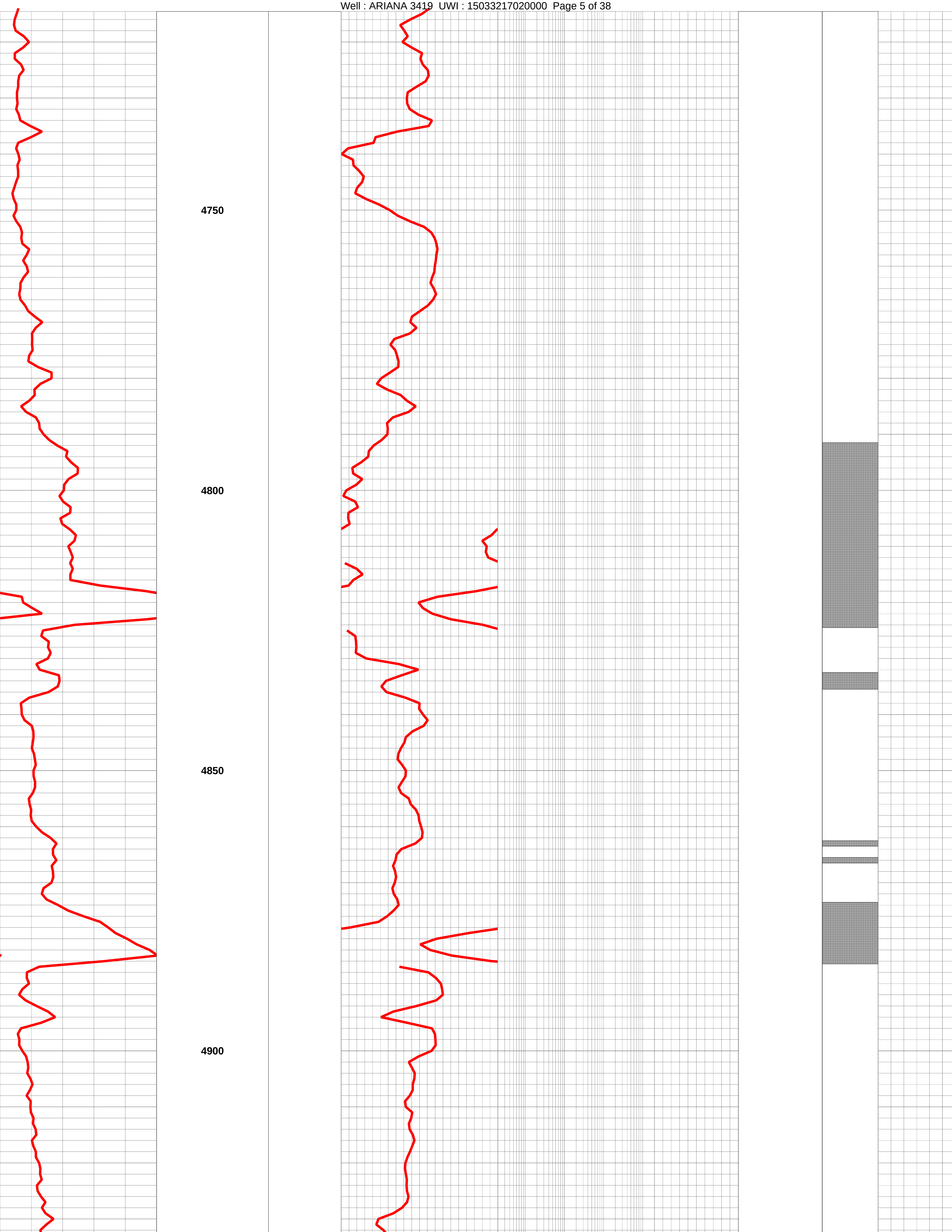
COMPUTATION			
	LOGS USED: Triple Combo		PROGRAM: PRIZM
	CENTER: OKC	LOG ANALYST: J. Austin	DATE: 05/10/2012
	REFERENCE NUMBER: [Modified Dual Water]		

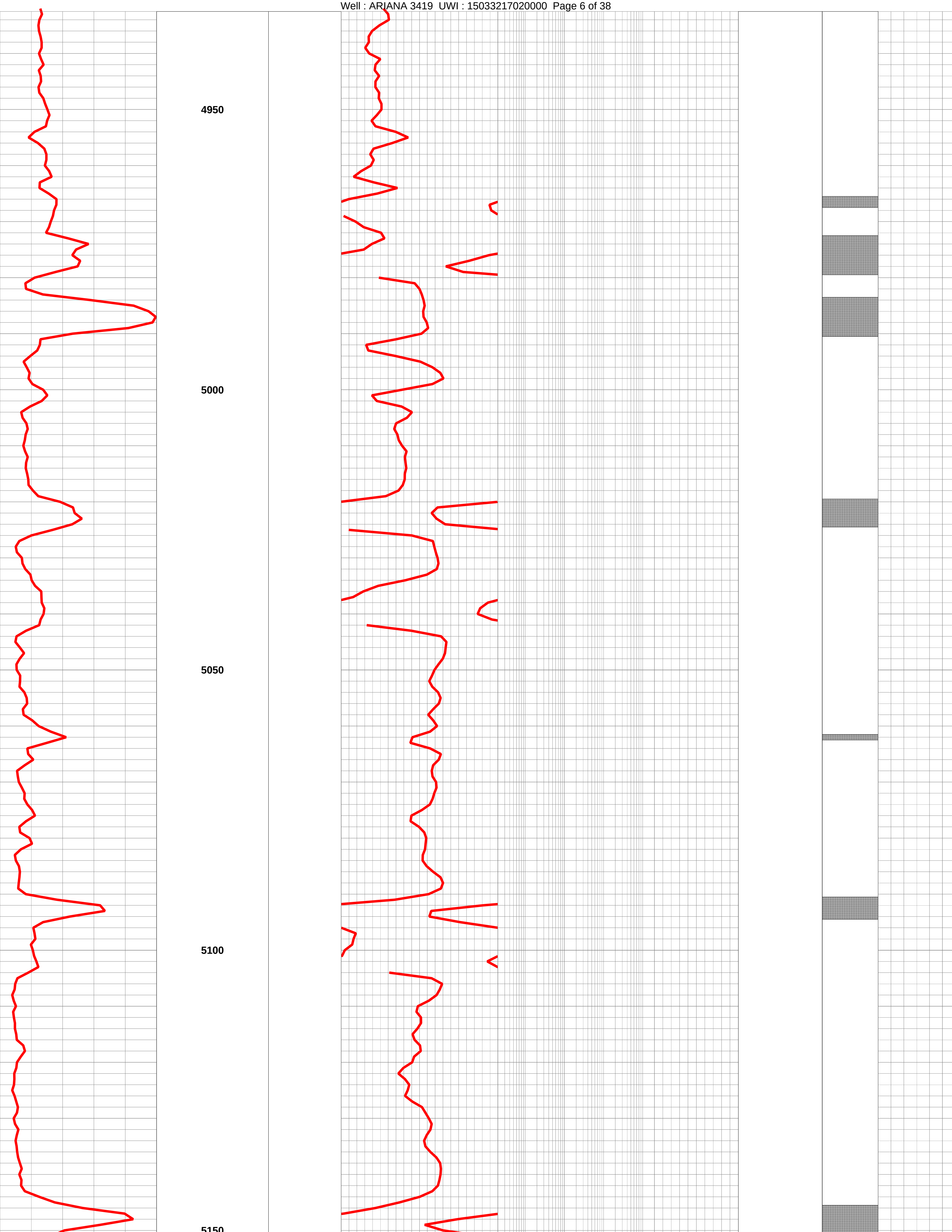


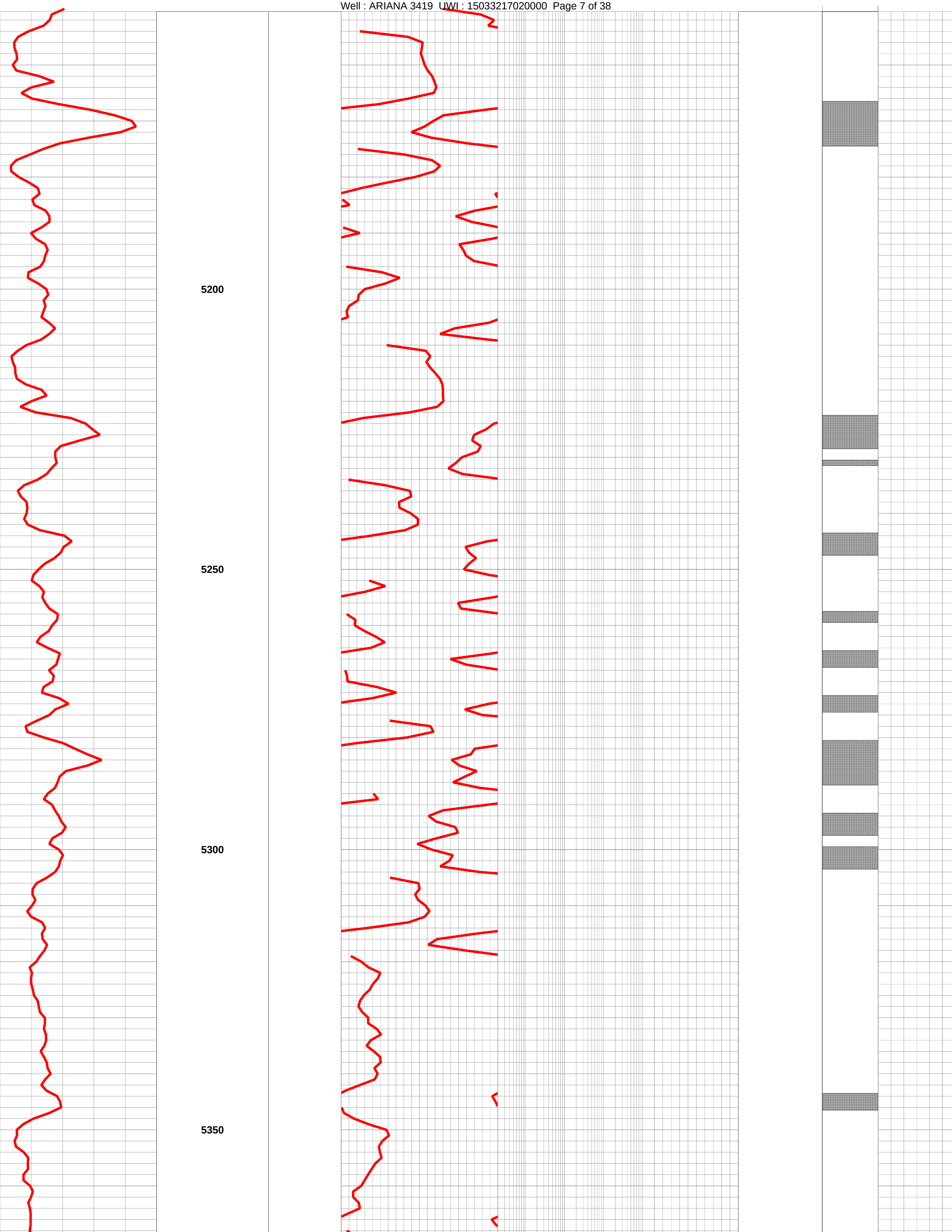

4100
4150
4200
4250


4300
4350
4400
4450

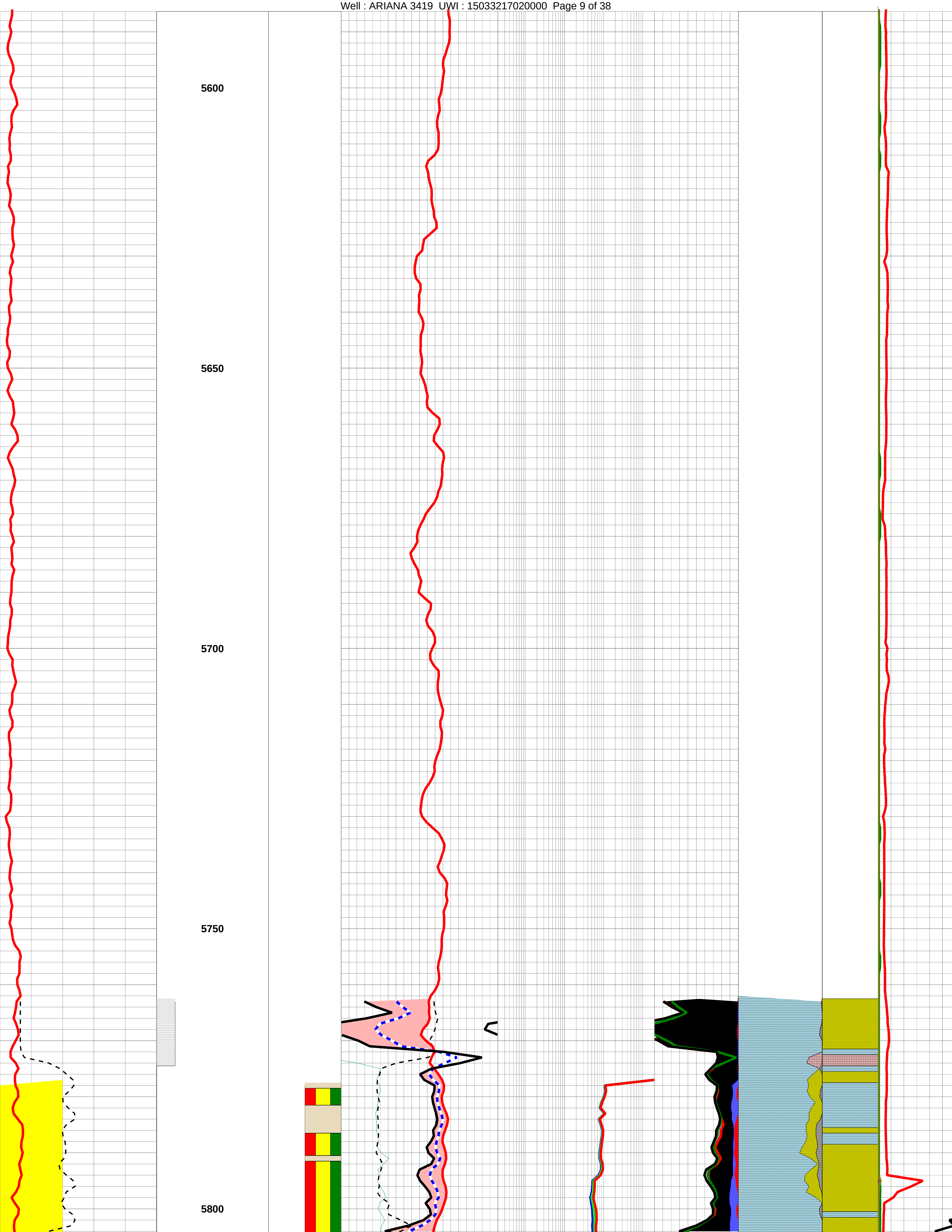

4500
4550
4600
4650
4700


4750
4800
4850
4900

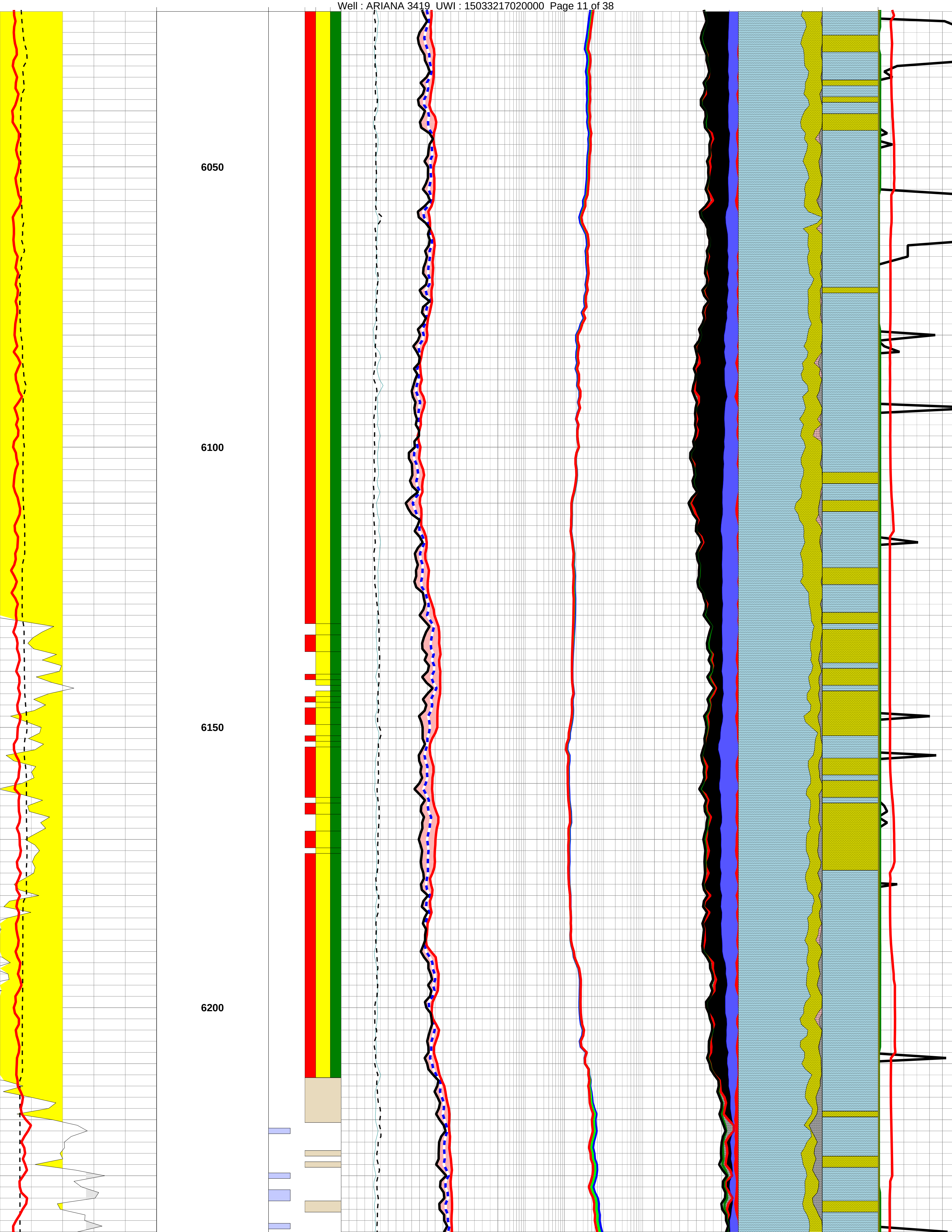

4950
5000
5050
5100
5150

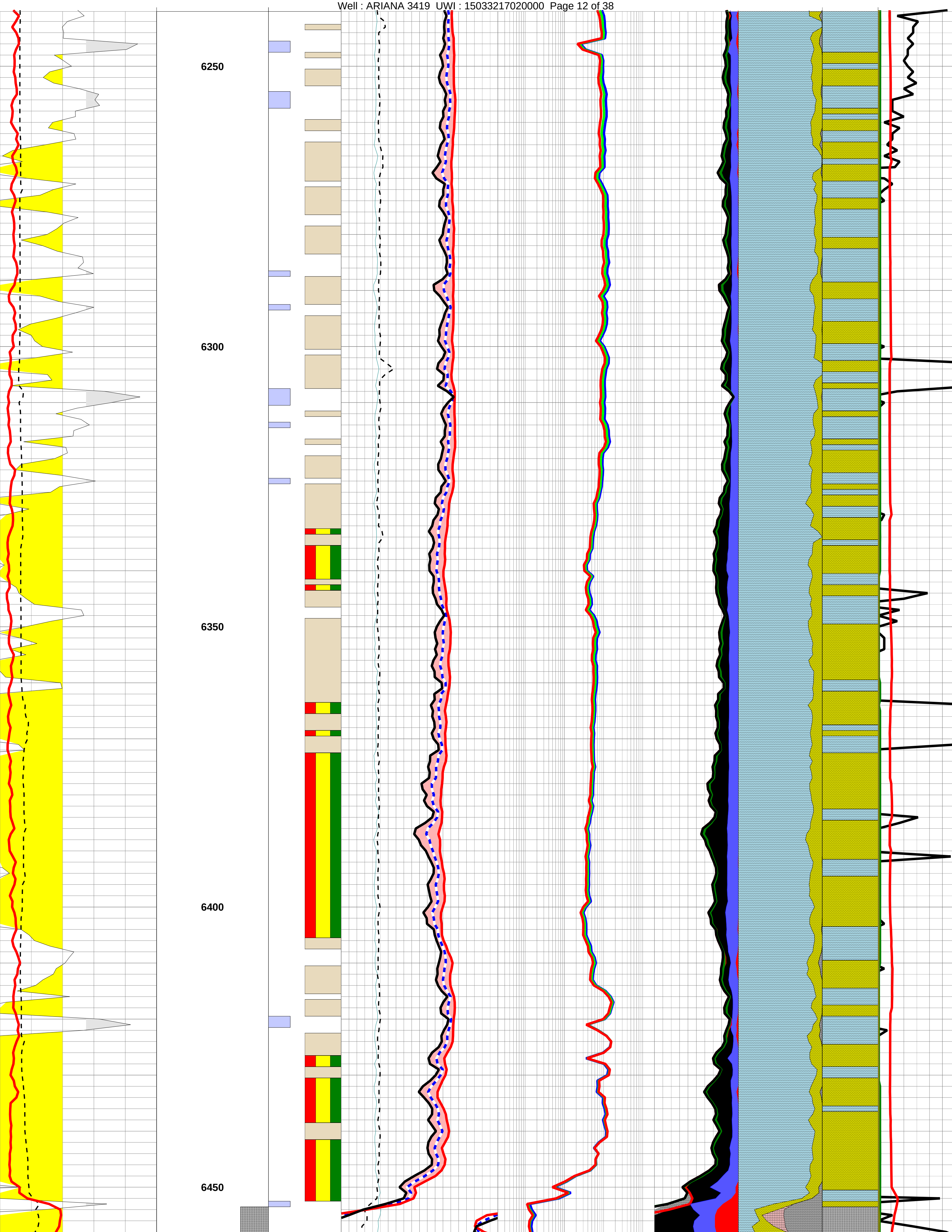

5200
5250
5300
5350

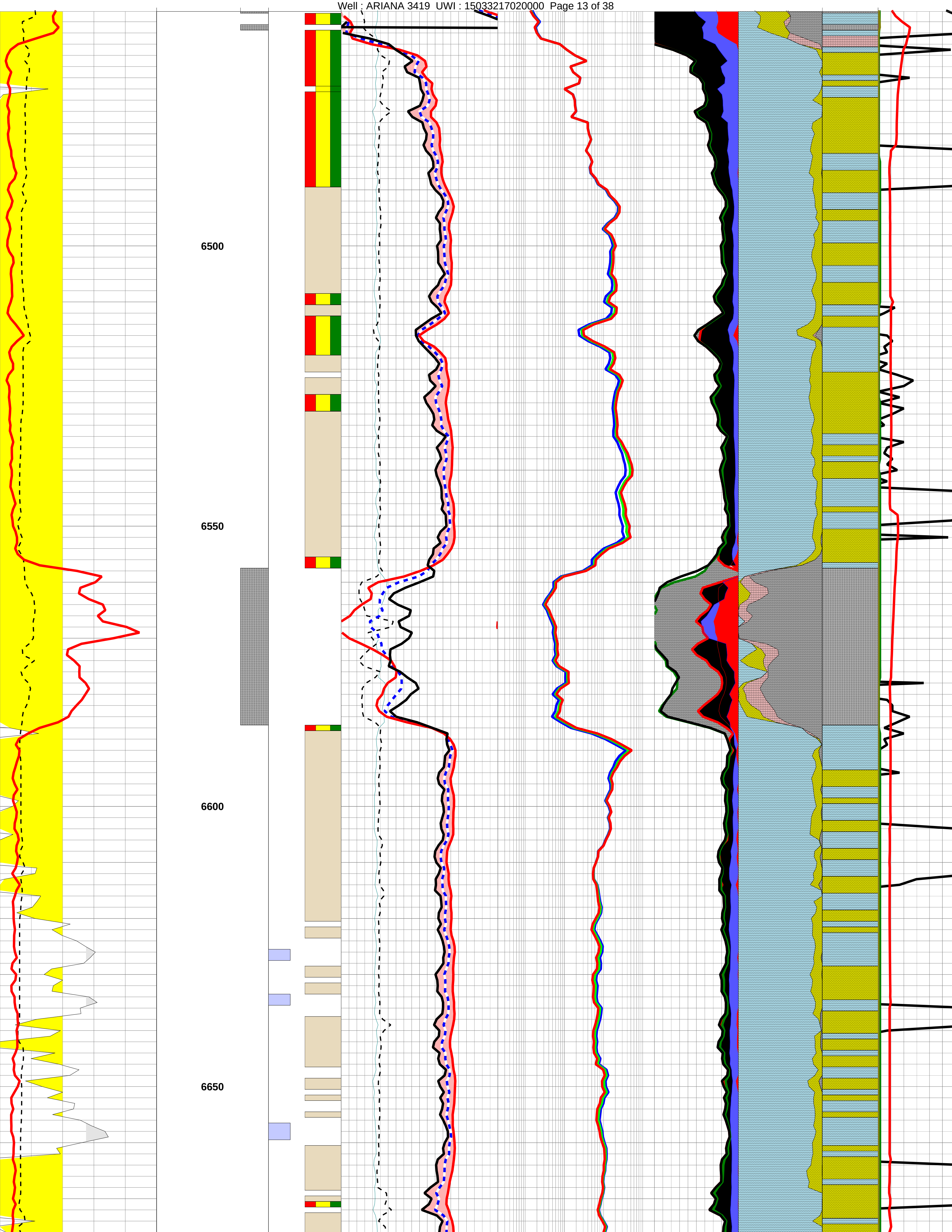

5400
5450
5500
5550

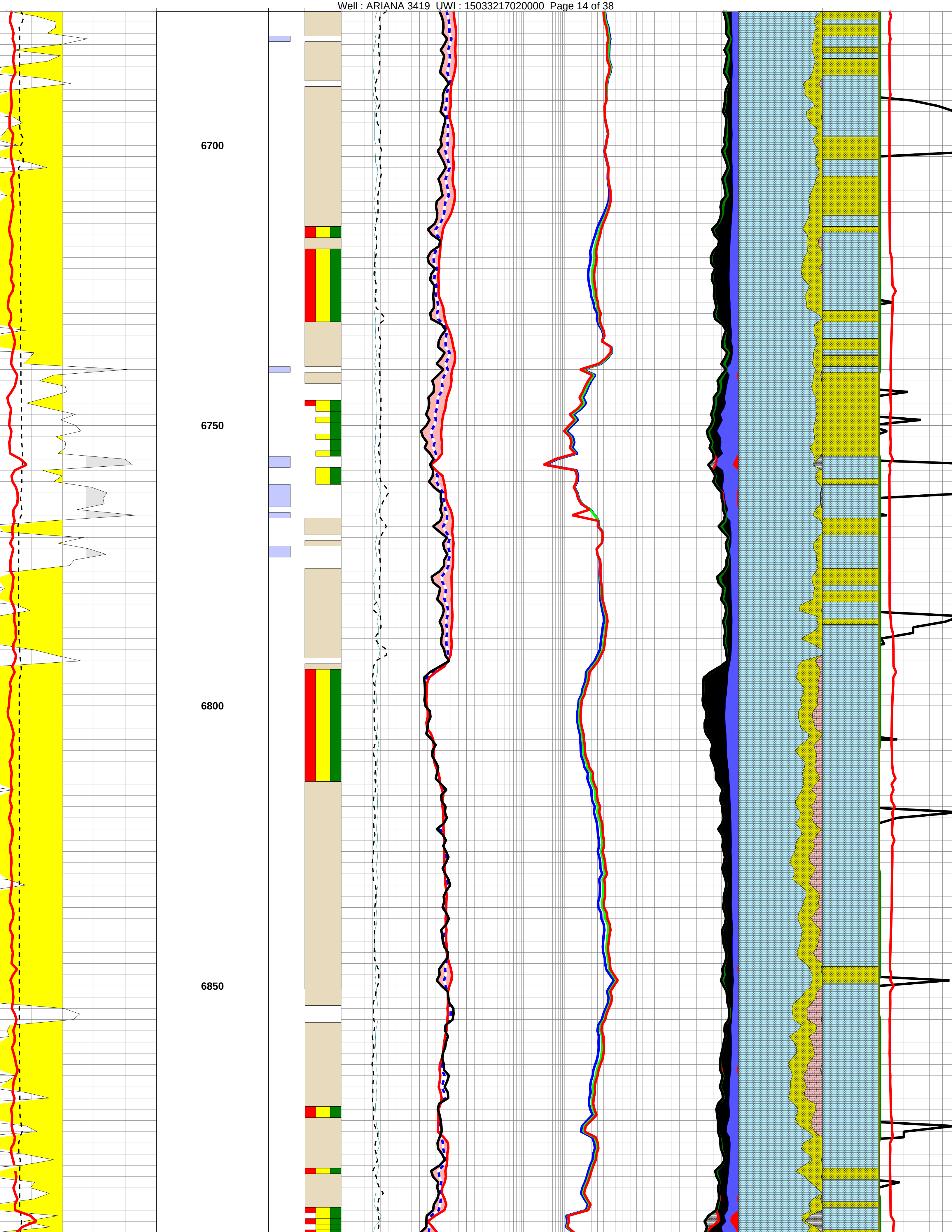

5600
5650
5700
5750
5800

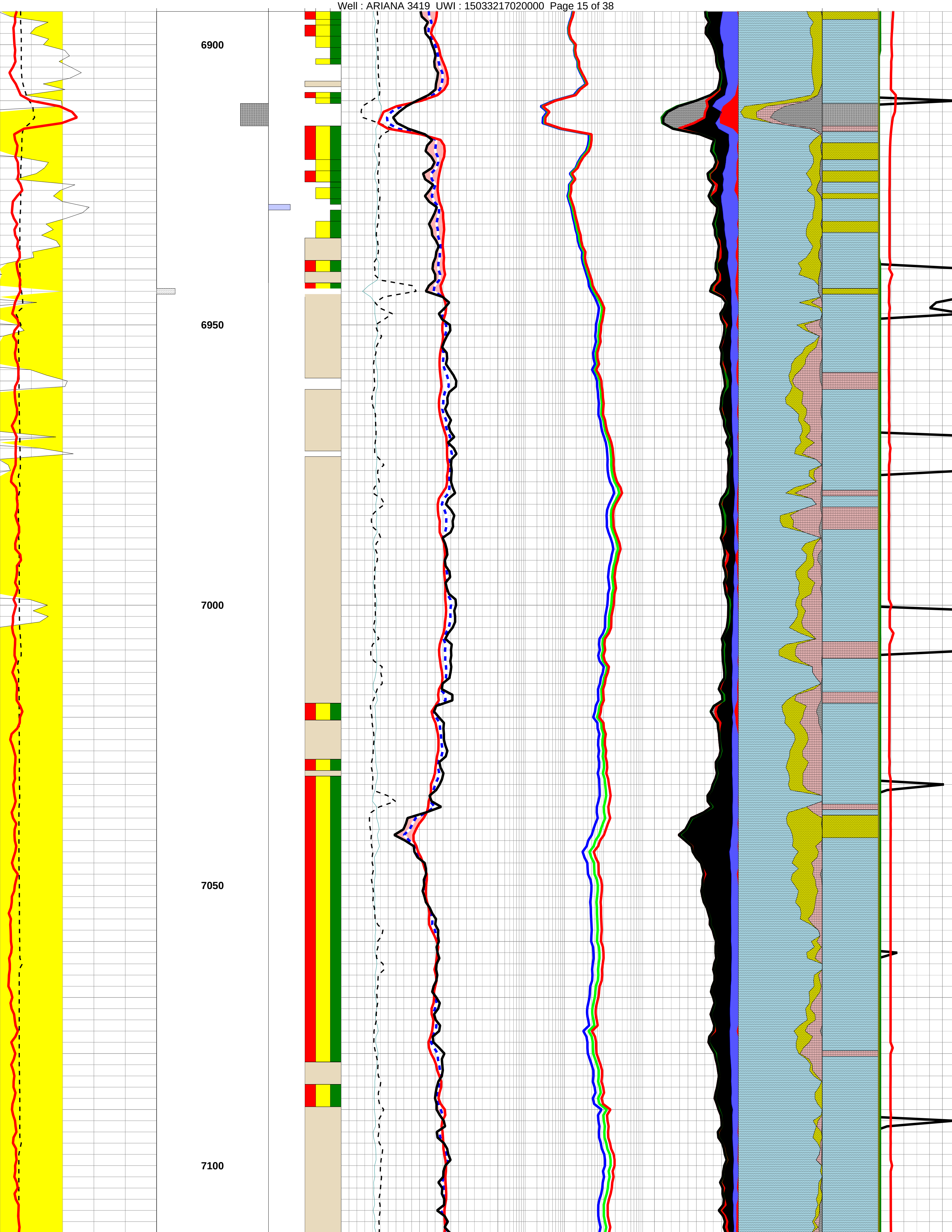

5850
5900
5950
6000

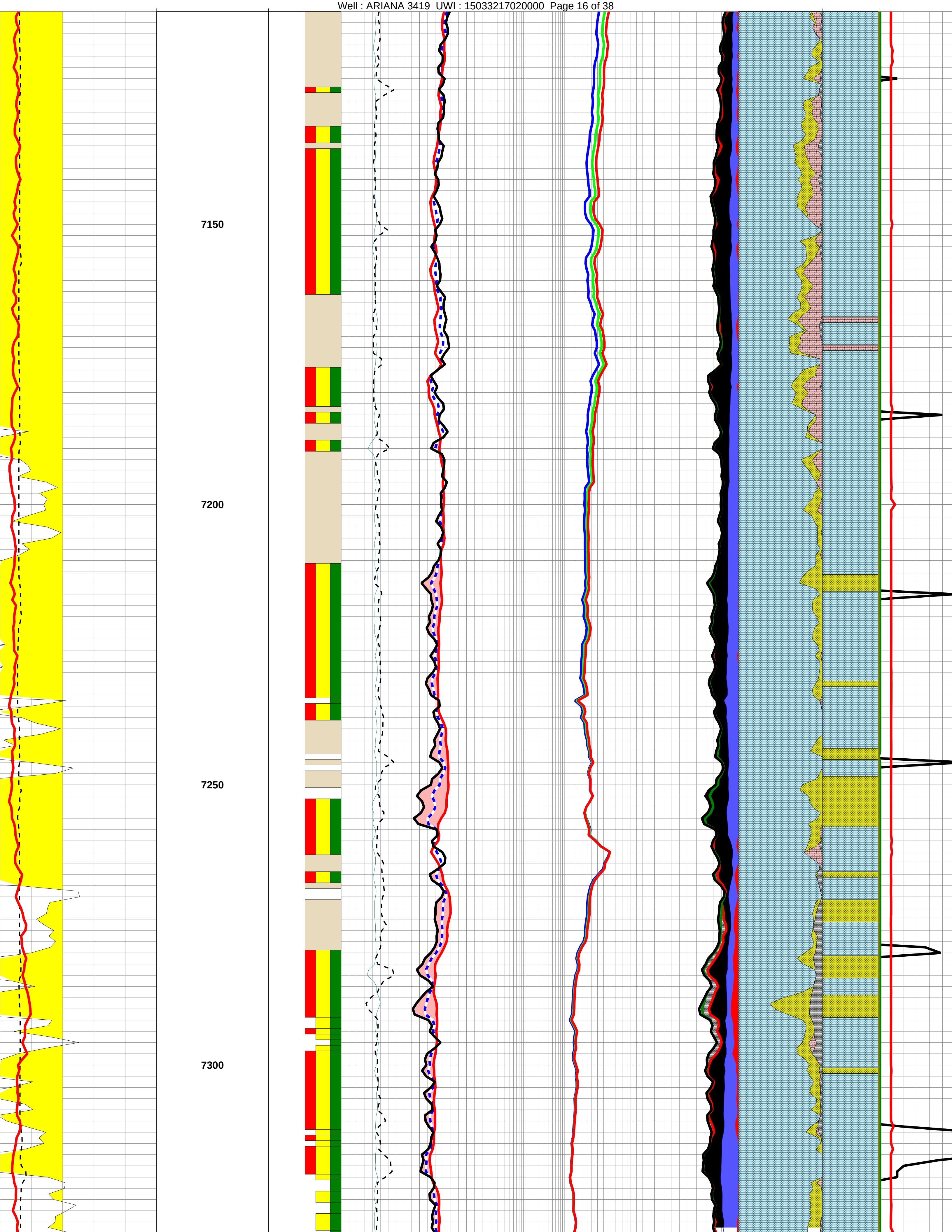

6050
6100
6150
6200

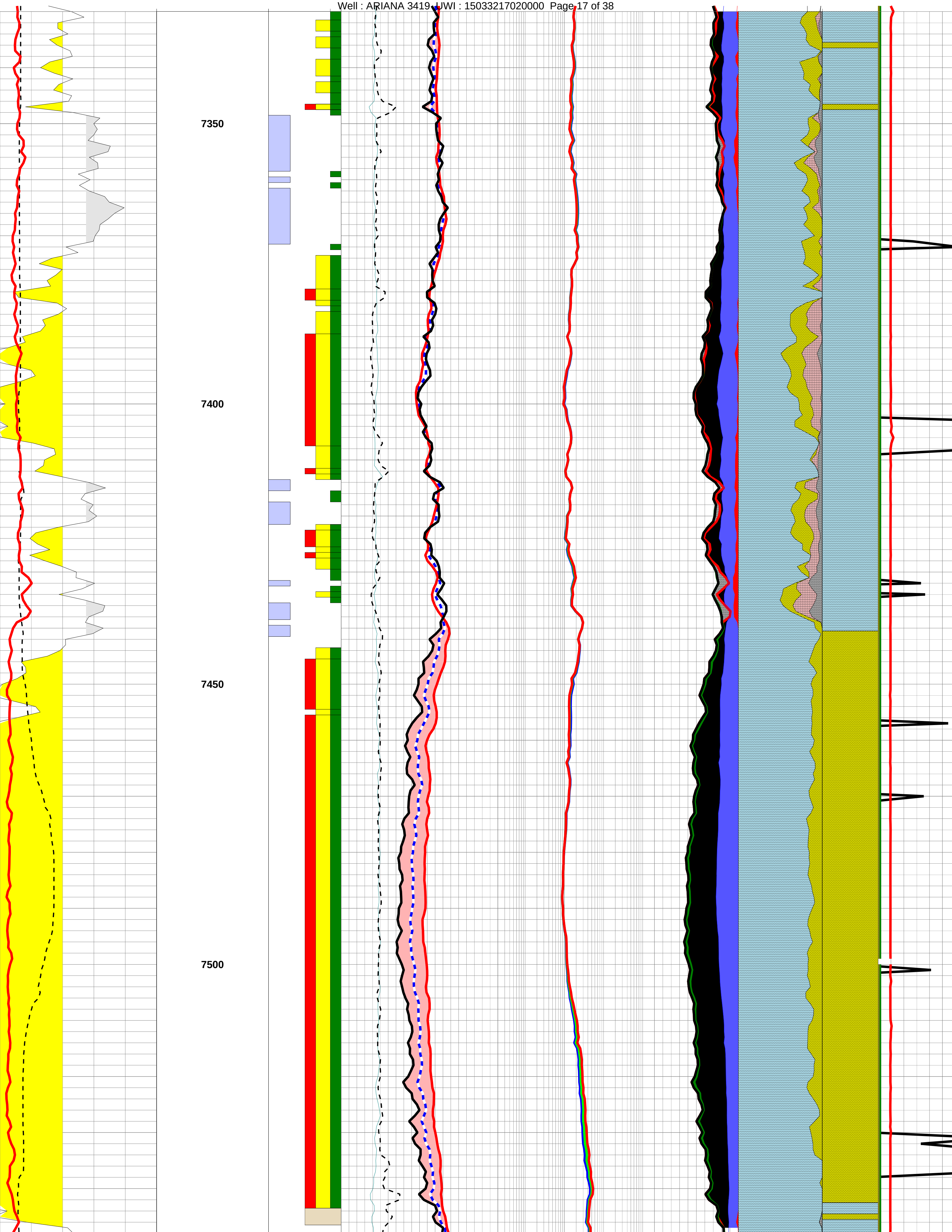

6250
6300
6350
6400
6450

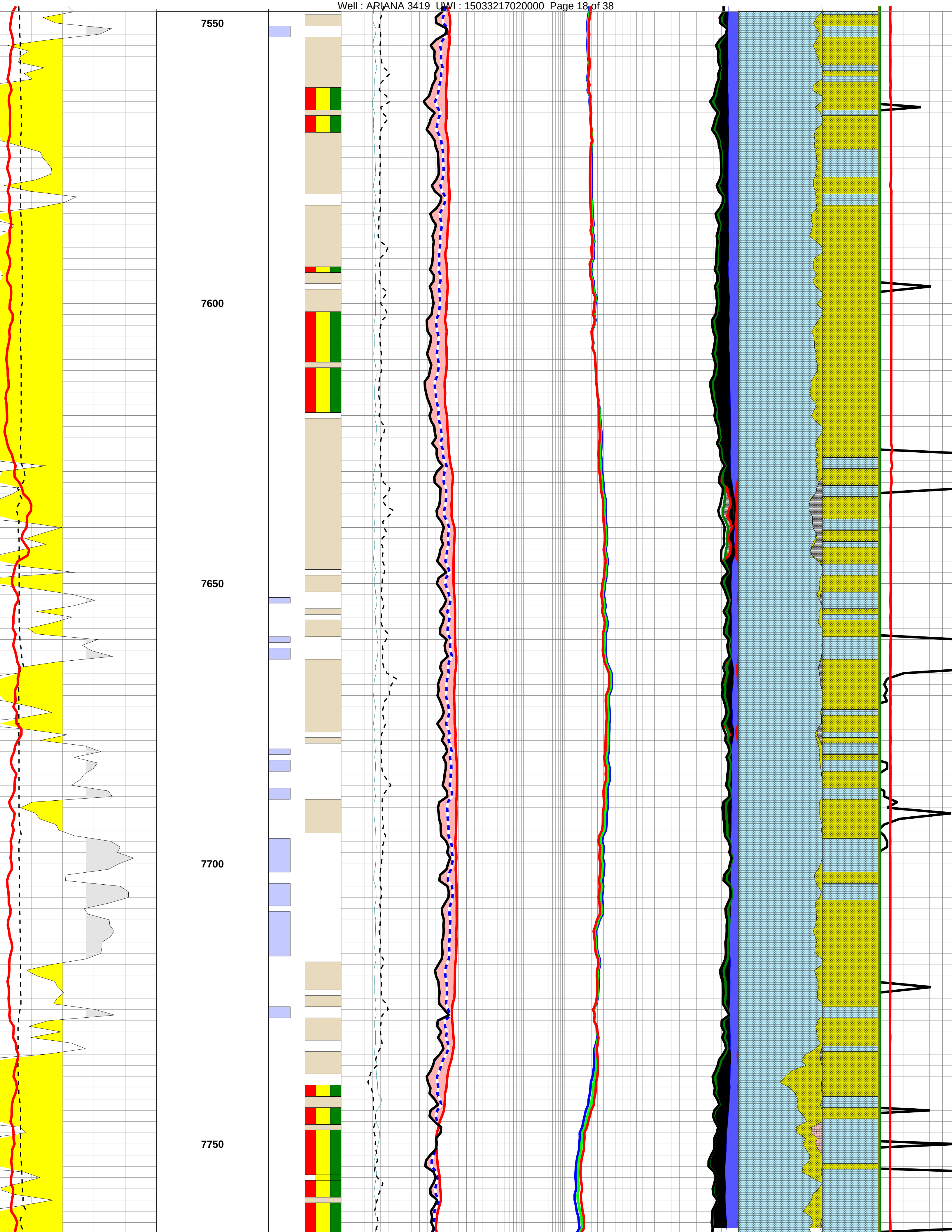

6500
6550
6600
6650

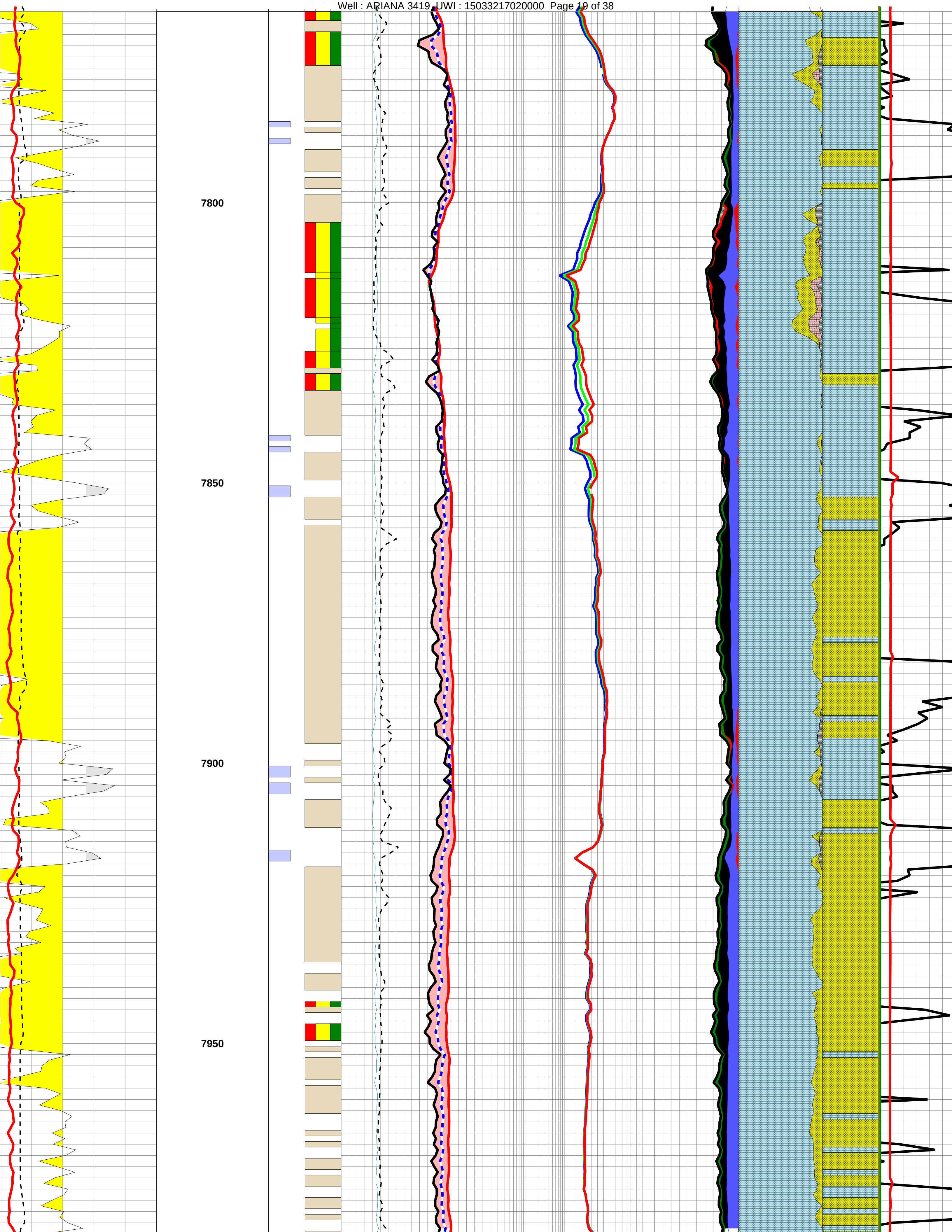

6700
6750
6800
6850

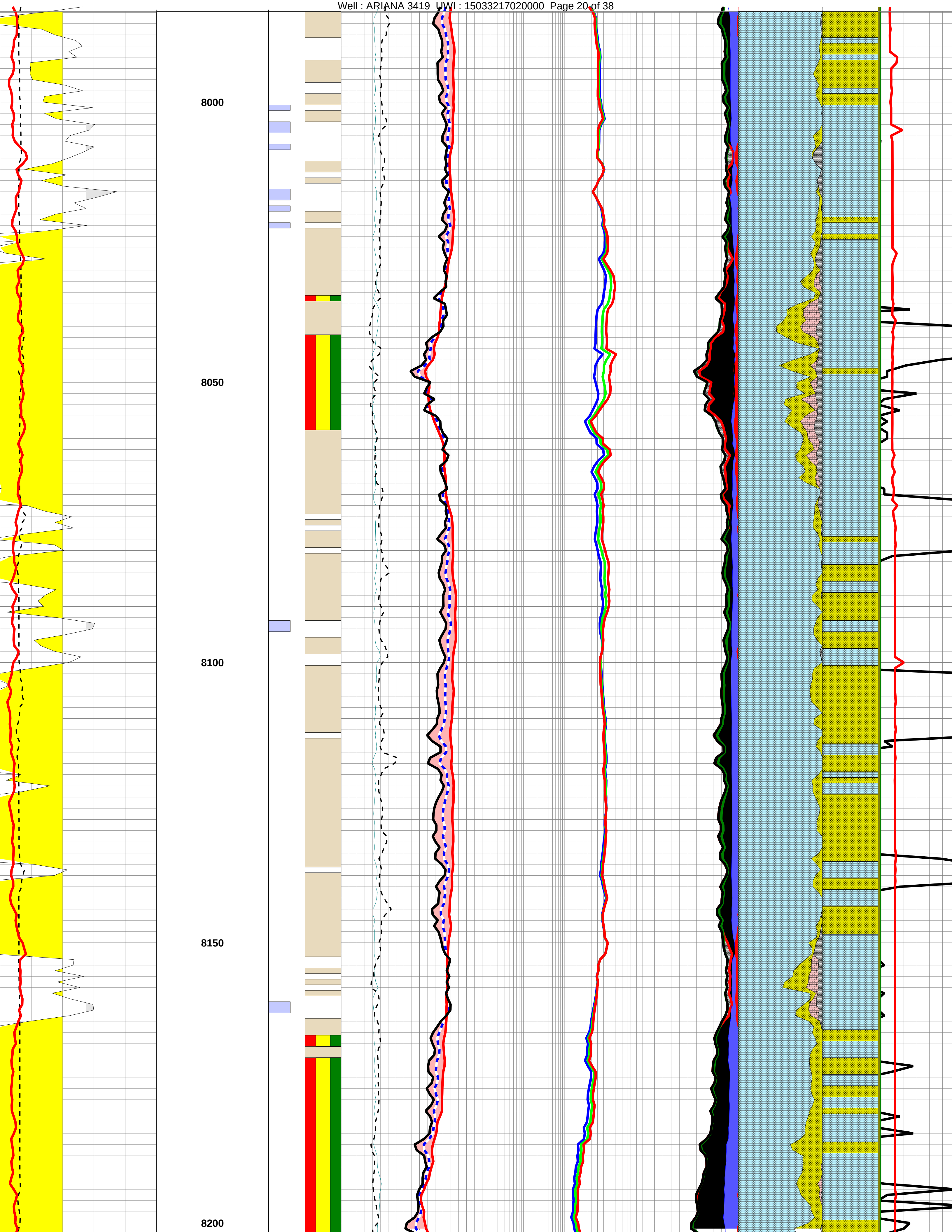

6900
6950
7000
7050
7100

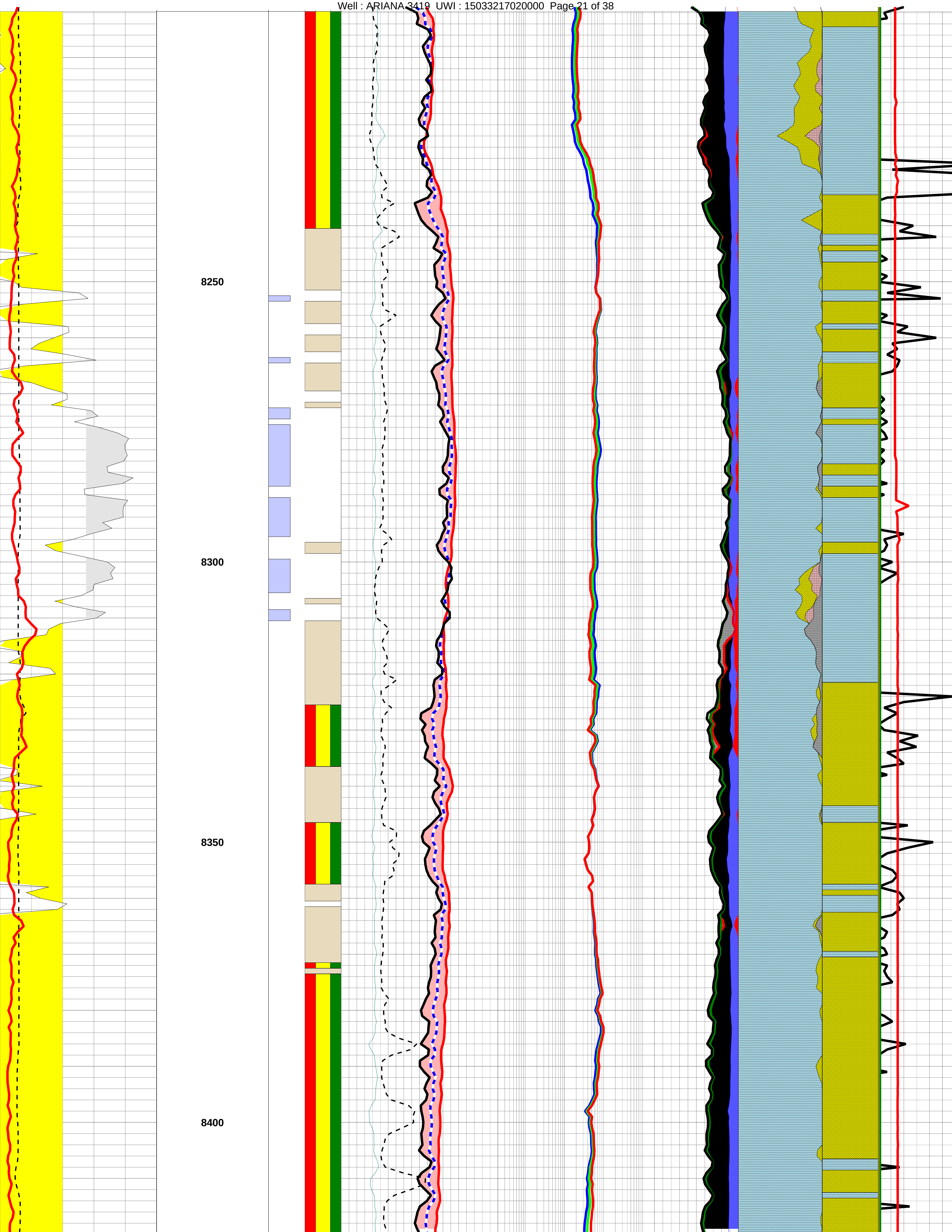

7150
7200
7250
7300

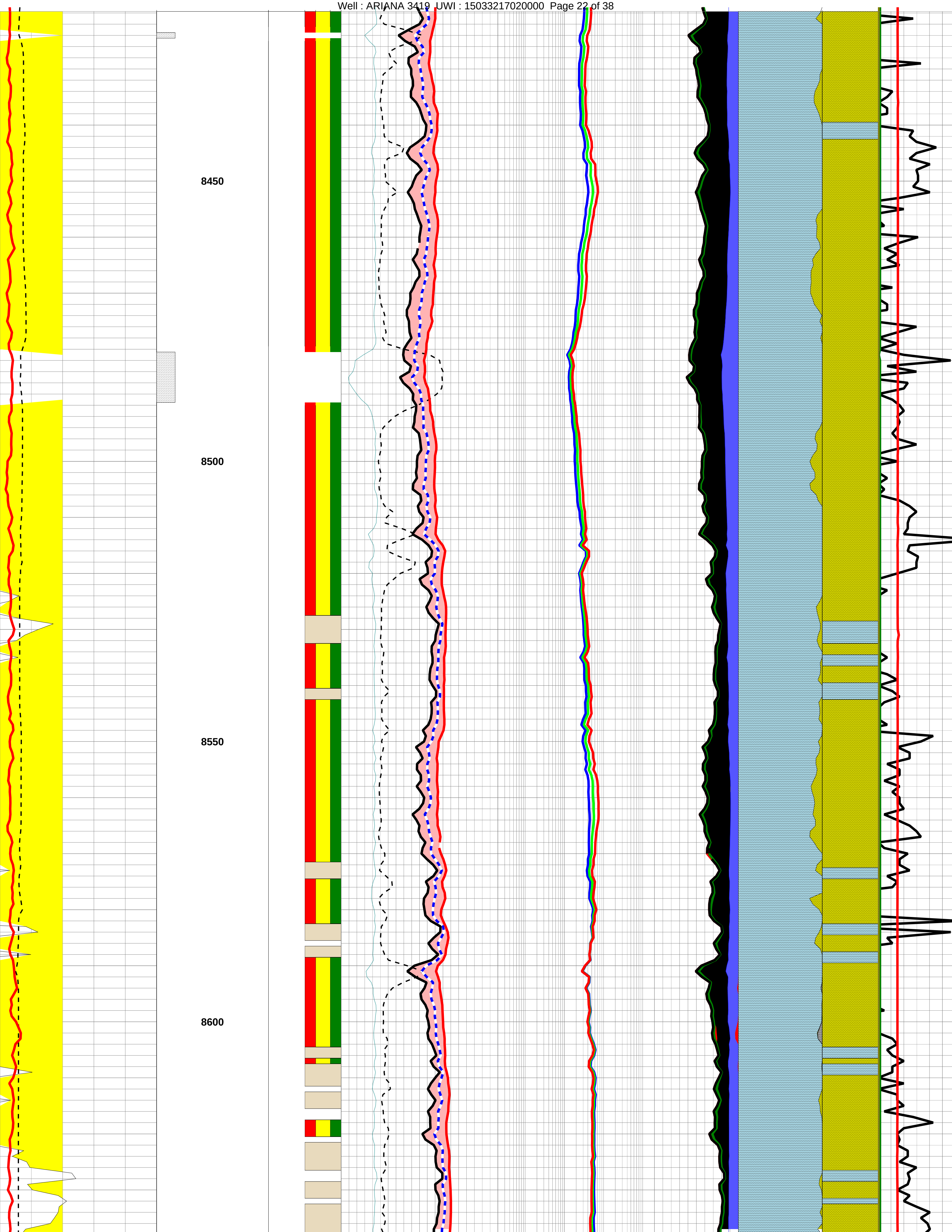

7350
7400
7450
7500

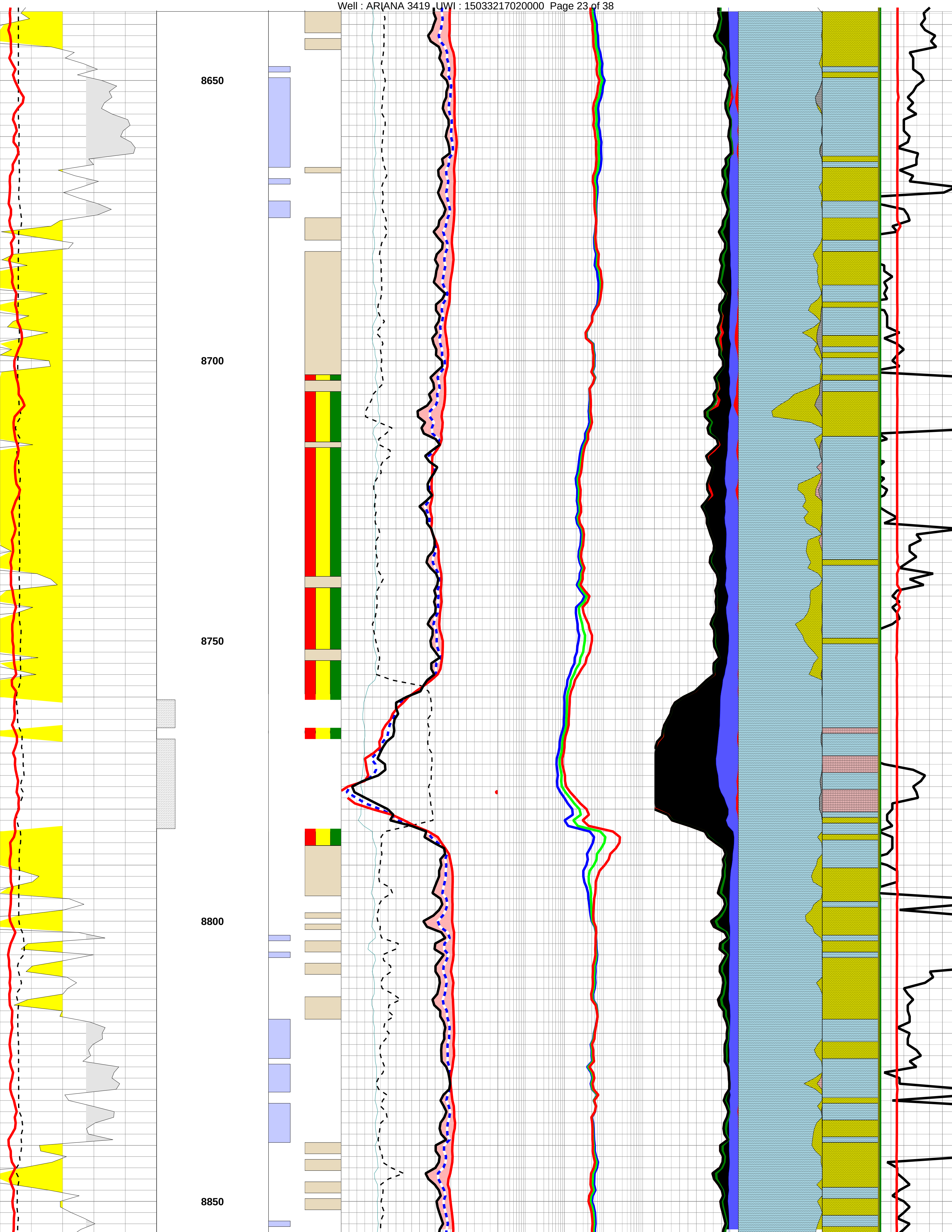

7550
7600
7650
7700
7750

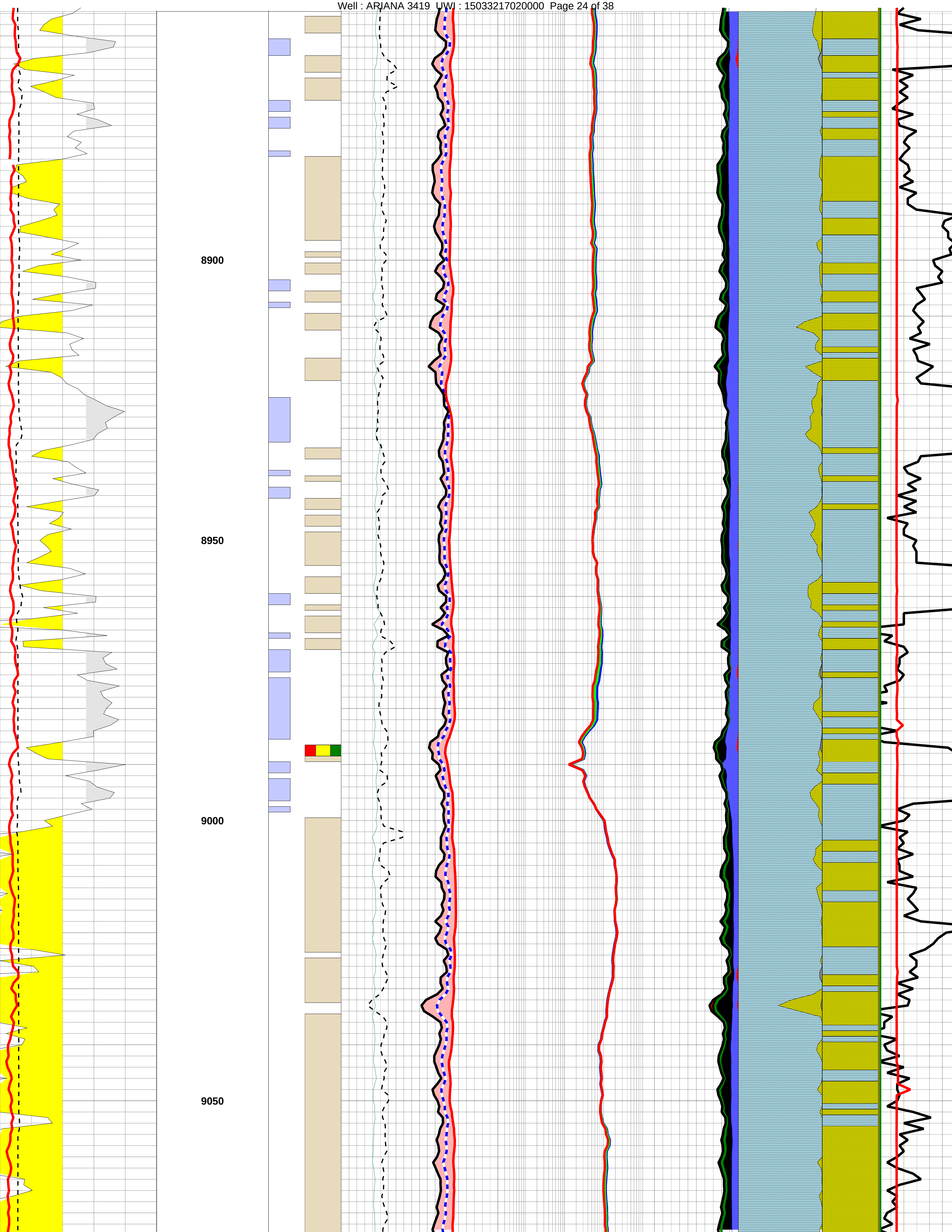

7800
7850
7900
7950

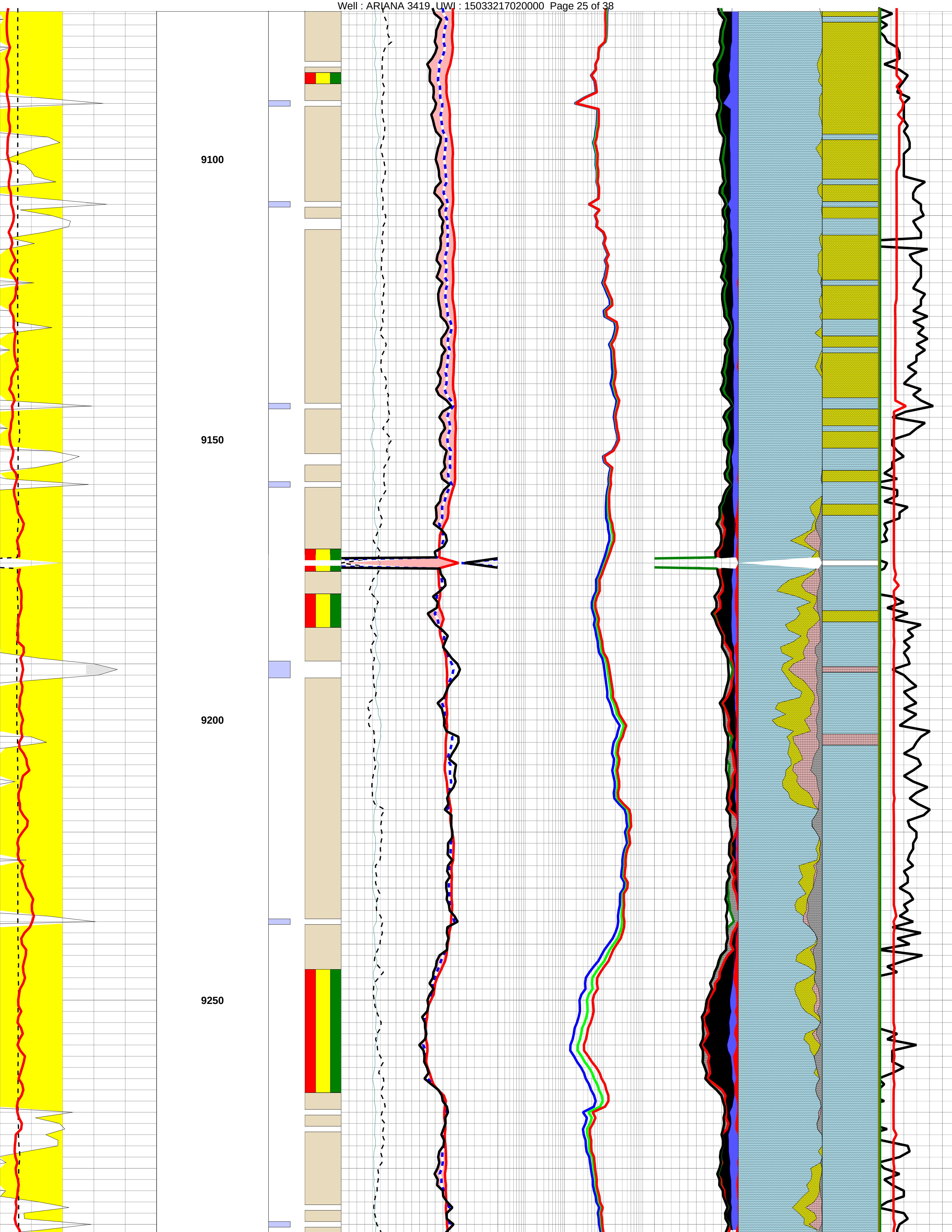

8000
8050
8100
8150
8200

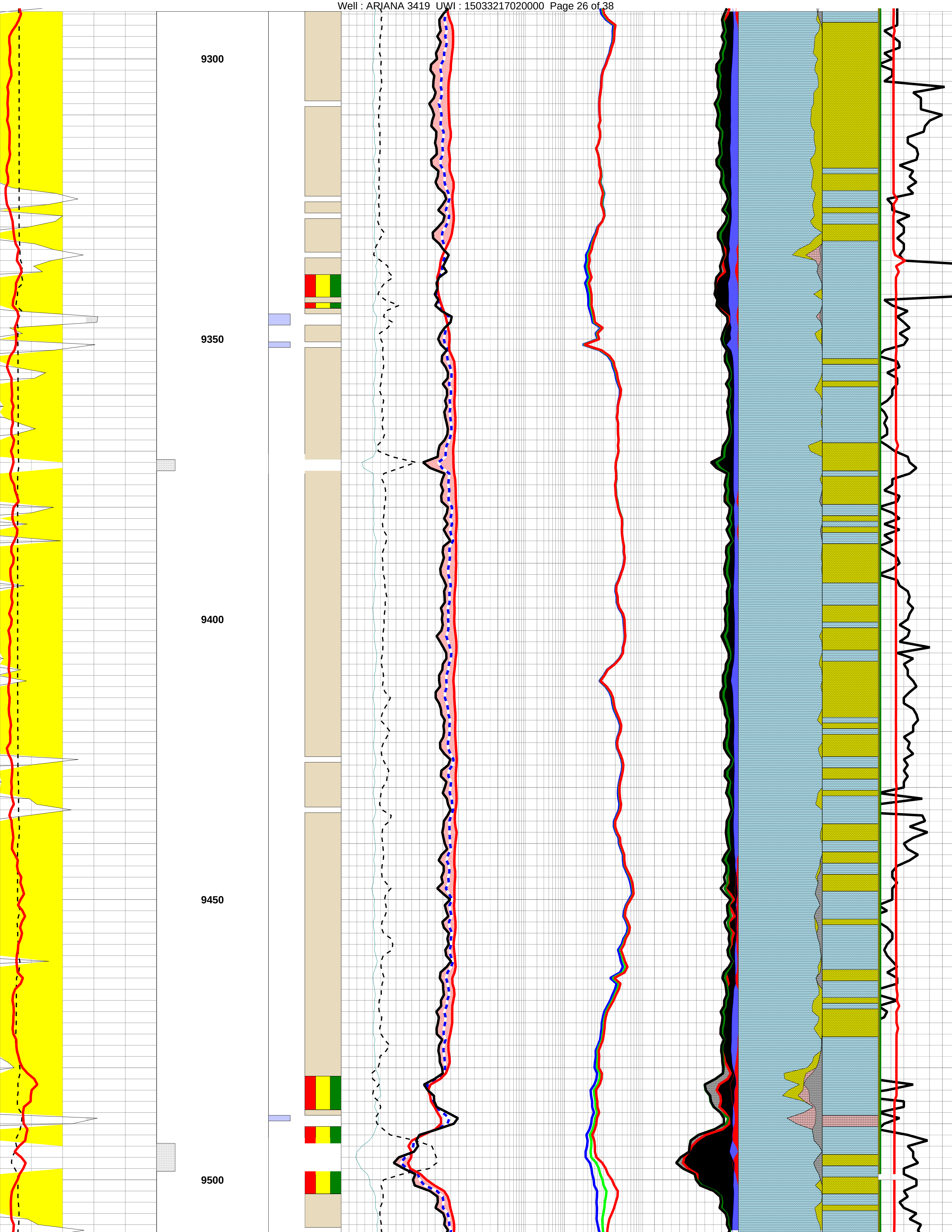

8250
8300
8350
8400

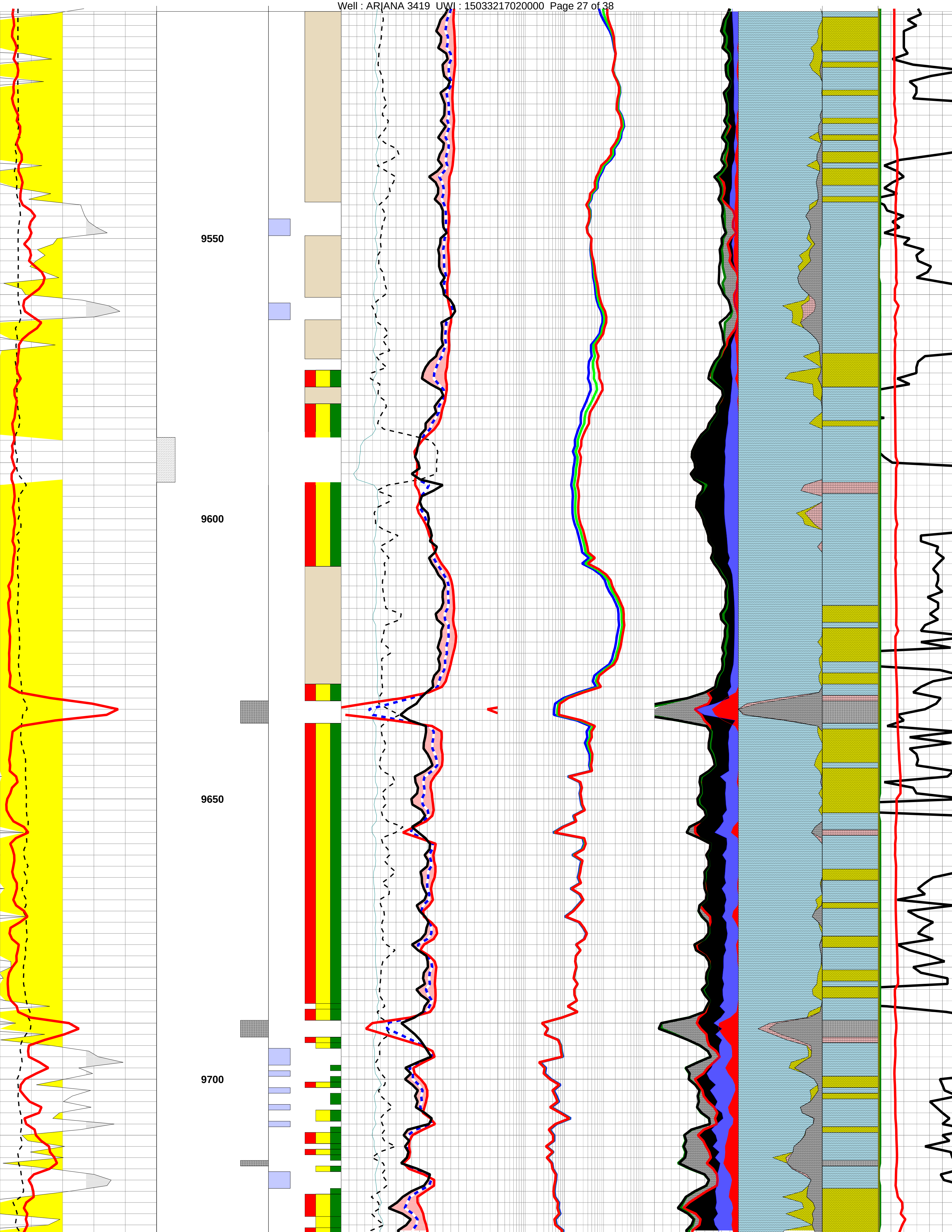

8450
8500
8550
8600

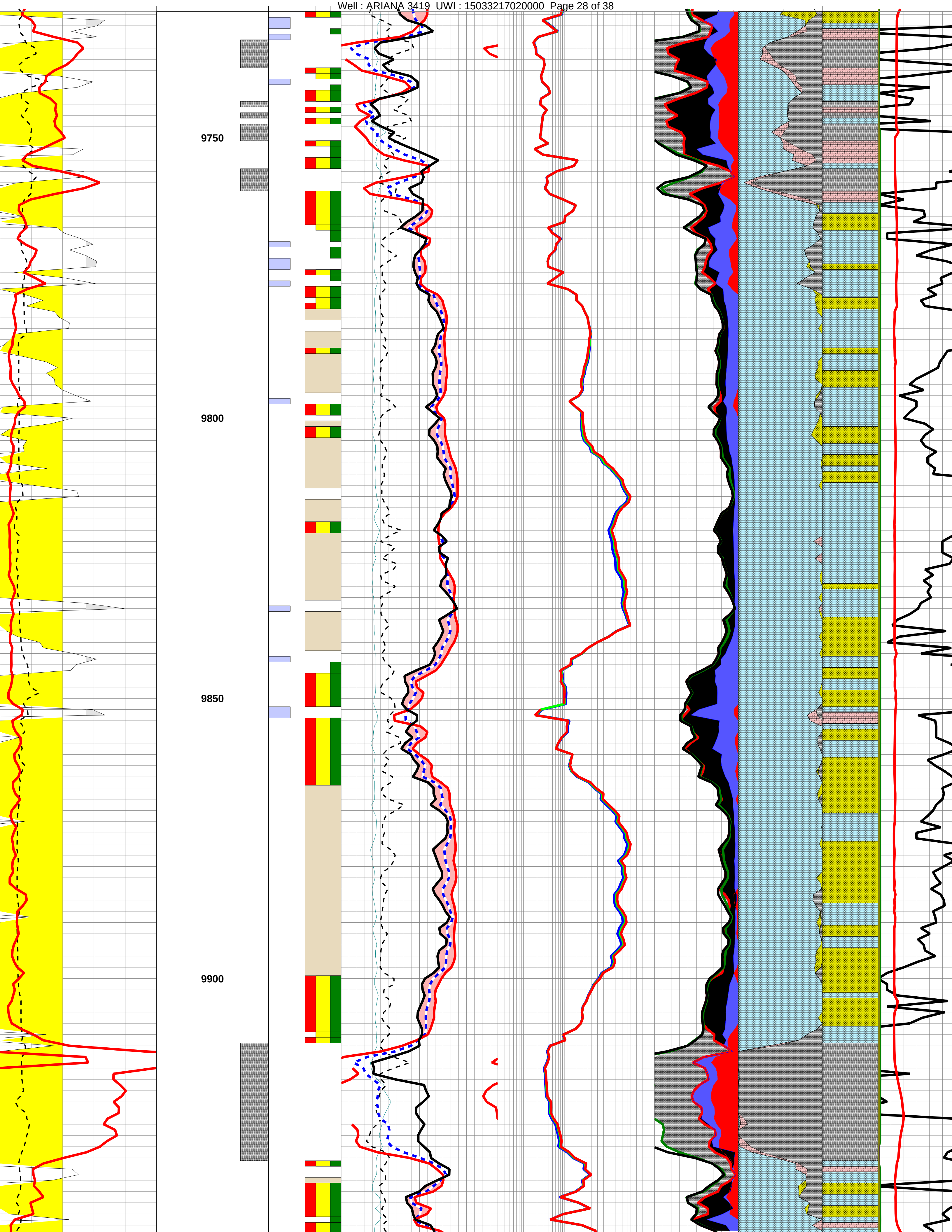

8650
8700
8750
8800
8850

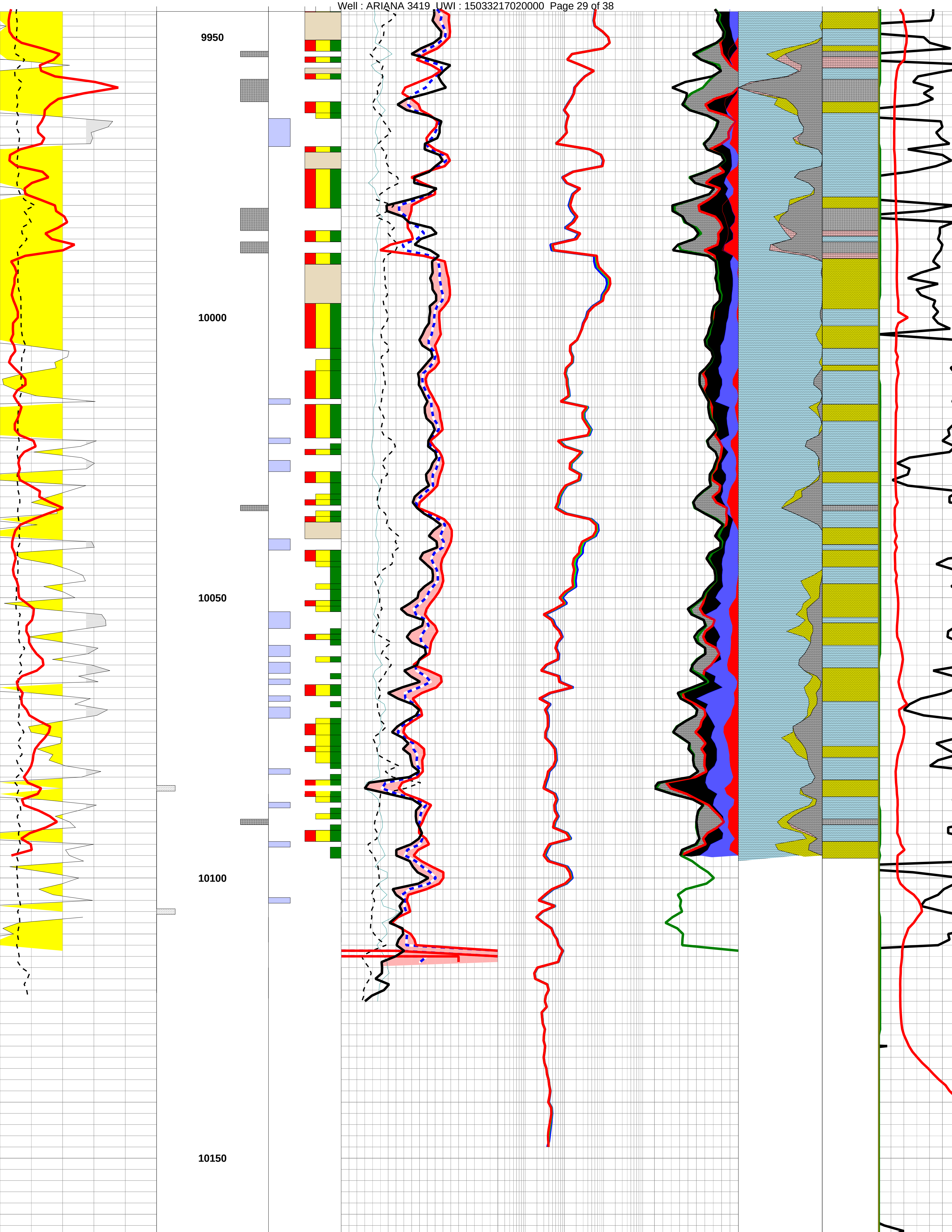

8900
8950
9000
9050


9100
9150
9200
9250


9300
9350
9400
9450
9500


9550
9600
9650
9700


9750
9800
9850
9900


9950
10000
10050
10100
10150

10200

10250

Correlation	Depth	RWA Flags	Open-Hole Porosity	Open-Hole Resistivity	Pore Space	Lithology	INT-LITH	Mudlog D
CGR(N/A) 0 150	MD	>=.075	PHID 0.3 -0.1	ResD(RTAO) 0.2 OHMM 2000.0000	PHIT 0.2 0	Sandstone	Sandstone	C1 0.0 units
GR(GGME) 0 GAPI 150	BHF(N/A)	>=.06	PHIN 0.3 -0.1	ResM(R60O) 0.2 OHMM 2000.0000	PHIA 0.20 0	Limestone	Limestone	C2 0.0 units
GRPO(N/A) 0 150	Vshl >.4(N/A)	>=.045	SXP 0.3 -0.1	ResS(R40O) 0.2 OHMM 2000.0000	BVW 0.2 0	Dolomite	Dolomite	C3 0.0 units
GRTH(N/A) 0 150	Vshl >.4	<.045	PEF(PDPE) 0.0 B/E 20		PHIE 0.20 0	Clay	Shale	C4 0.0 units
GRUR(N/A) 0 150		Tight Pay	DRHO(DCOR) -0.25 G/C3 0.75		BVWb 0.2 0			OIL_CU 0.0
CALI(CLDC) 5 IN 15			Crossover		Hydrocarbon			ROP 150
RWA_SXP 0.1 0.0					Bulk Volume Water			
RWA<.045					Clay Water			
RWA>=.06					Bound Water			

User Defined Equations
; Dual Water 4 Mineral Model - GAS ZONES
; Input Curves: GR, RHOB, NPHI, RT, PEF
; Revision: July 10, 2012
; Jack W. Austin after G. B. Asquith, AAPG CEC Series #31

; Mud Log Sample Percentates for Display
SHALE_disp[] = Shale[]
CarbSh_disp[] = SHALE_disp[] + Carb_shale[]
SiltST_disp[] = CarbSh_disp[] + SiltStone[]
SS_disp[] = SiltST_disp[] + Sand[]
LMST_disp[] = SS_disp[] + Lime[]
DOLO_disp[] = LMST_disp[] + DOLOMITE[]
Chert_disp[] = DOLO_disp[] + Chert[]
;Anhy_disp[] = Chert_disp[] + ANHY[]

; Bad Hole Flag
BHF[] = CALI[] > 10 or DRHO[] > 0.10 or DRHO[] < -0.10

```
;              Calculated Gamma Ray from Spectral comparison
;CGR[] = GRDI[] * 0.14 + 15

;              SHALE (CLAY) VOLUME
;Igr[] = (GR[]-GRcln) / (GRshl-GRcln)
Vshl[] = min(1, max(0, (GR[]-GRcln) / (GRshl-GRcln) ))
;Vshl[] = min(1, max(0, 0.33 * (2^(2 * Igr[]) - 1.0) ))
;Vshl[] =  min(1, max(0, (RHOB[] - RhoM + PHIN[] * (RhoM - RhoF)) / (RhoShl - RhoM +HIshl * (RhoM - RhoF)) ))

;              Sonic Porosity
SPHI[] = (DT[] -DTma) / (DTfld - DTma)

;              DENSITY POROSITY
If (IsNull(RHOB[]))
PHID[] = PHID[]
Else
;if (IsNull(DGA[]))
PHID[] = (RhoM - RHOB[]) / (RhoM - RhoF)
;Else
;PHID[] = (DGA[] - RHOB[]) / (DGA[] - RhoF)
End If

;              Matix Corrected Neutron Porosity
;   Limestone to Dolomite
;   SWS_CNL_Matrix_LStoDM (Por-13a)
PHIN_DMs[] = SWS_CNL_Matrix_LStoDM(PHIN[])
;   BA_2446CN_LStoDM
PHIN_DMb[] = BA_2446CN_LStoDM(PHIN[], 1)
;   HES_DSN_LStoDM (POR-12 DSN-II)
PHIN_DMh[] = PHIN[] + HES_DSN_LStoDM(PHIN[])
If (PhiN_DM = 0) Then PHIN[] = PHIN[]
If (PhiN_DM = 1) Then PHIN[] = PHIN_DMs[]
If (PhiN_DM = 2) Then PHIN[] = PHIN_DMb[]
If (PhiN_DM = 3) Then PHIN[] = PHIN_DMh[]

;              SHALE CORRECTED POROSITY
PHIDc[] = max(0,PHID[] - (Vshl[] * PhiDshl) )
PHINc[] = max(0,PHIN[] - (Vshl[] * PhiNshl) )

;              AVERAGE POROSITY - ARCHIE
If (IsNull (RHOB[])) Then PHIA[] = SPHI[] Else PHIA[] = (PHID[] + PHIN[]) / 2

;              Cross Plot Porosity
;  SWS CP-1e RhoF = 1.0 g/cc
PHIXs[] =SWS_LDT_CNL_CP1e_E( PHIN[], RHOB[] )
;  BA Por & Lith 2446CN RhoF = 1.0 g/cc
PHIXb[] = BA_CD_2446CN_XPor_Fresh_E( PHIN[], RHOB[] )
;  HES CPdsnII-1b RhoF = 1.0 g/cc
PHIXh[] = HES_SDL_DSN_Porosity_1E( PHIN[], RHOB[] )
If (XPlotPor = 1) Then PHIXp[] = PHIXs[]
If (XPlotPor = 2) Then PHIXp[] = PHIXb[]
If (XPlotPor = 3) Then PHIXp[] = PHIXh[]

;******************SXP**************************************************************************************************
;Simple Cross Plot Porosity
SXP[] = (PHID[] + PHIN[])/2
;*****************End SXP***********************************************************************************************

;Interpreted Lithology Display based upon porosity curves
SS_fill[] = 0
DOL_fill[] = 0
LIM_fill[] = 0
SHL_fill[] = 0
PDIFF[] = PHID[] - PHIN[]
If (Vshl[] <= .4 and PDIFF[] >= .03) Then (SS_fill[] = 10)
If (Vshl[] <= .4 and PDIFF[] <= -.03) Then (DOL_fill[] = 10)
If (Vshl[] <=.4 and abs(PDIFF[]) <.03) Then (LIM_fill[] = 10)
If (Vshl[] >.4) Then (SHL_fill[] = 10)
;              TOTAL POROSITY ADJACENT SHALE
PHITsh = (d * PhiDshl) + ((1-d) * PhiNshl)

;              EFFECTIVE POROSITY
;                             GAS
; Original equations DO NOT contain SXP[], the modified equations DO
;The line below is a shale corrected Effective Porosity
PHIE[] = ((PHINc[]^2 + PHIDc[]^2) / 2)^0.5
;PHIE[] = SXP[]
;The line below is a shale corrected and tool corrected Effective Porosity
;PHIEx[] = PHIXp[] * (1-Vshl[])
;The line below takes out the tool correction
PHIEx[] = SXP[] * (1-Vshl[])
;for area fill purposes...in clean rock PHIA is nearly equal to PHIT.  JA had it set up so that clay water was an
;area fill between PHIA and PHIE, I've had to change it so that it is from PHIT to PHIE using SXP[].  This clay water
;is way overstated then in shaley intervals.

;                             OIL
;If (BHF[] < 1) Then PHIE[] = ((PHINc[] + PHIDc[]) / 2) Else PHIE[] = 0

;              TOTAL POROSITY
PHIT[] = PHIE[] + (Vshl[] * PHITsh)
PHITx[] = PHIEx[] + (Vshl[] * PHITsh)

;              BOUND WATER
Swb[] = Vshl[] * PHITsh / PHIT[]
Swbx[] = Vshl[] * PHITsh / PHITx[]

;              RESISTIVITY of BOUND WATER and CLAY
Rwb = Rshl * PHITsh
Rcl = 0.4 * Rshl

;              APPERENT WATER RESISTIVITY
Rwa[] = RT[] * PHIT[]^m
Rwax[] = RT[] * PHITx[]^m

;******************Calculated Oil Curve From Gas Data*****************************************************************
```

```
OIL_CURVE[] = ((C3[]+C4[])/C1[])
;********************End of Calculated Oil Curve From Gas Data**************************************************


;*********************ROP Conversion to Feet Per Hour**************************************************************
ROP2[] = 60/ROP[]
;*********************End of ROP Conversion*************************************************************************


;*****************RWA**************************************************************************************************

;Make sure the logging tools are not inside casing or in Bad Hole
If ((ISNULL(DRHO[])) and (ISNULL(CALI[])))
RWA_SXP[] = <NULL>
Else
If (BHF[])
RWA_SXP[] = <NULL>
Else


;Apparent Resistivity Calculation
;RWA_SXP[] = (SXP[]^2) * ResD[]
RWA_SXP[] = (SXP[]^2) * RT[]

;Check for Bad Rwa Data
If (RWA_SXP[]  >= 5)
RWA_SXP[] = <NULL>
Else
If (RWA_SXP[] = 0)
RWA_SXP[] = <NULL>
Else

;RWA_FLAG Calculation

x[]=PHIN[] >= .03 and SXP[] >= .05 and Vshl[] <= .4
;Good
If( x[] and RWA_SXP[] >=.045)
RWA_LOW[] = 3
RWA_MID[] = 0
RWA_HI[] = 0
Pay_Type1[] = 1
Cum_Pay[]=1
Else
RWA_LOW[] = 0
Pay_Type1[] =0
Cum_Pay[]=0
Endif

;Better
If (x[] and RWA_SXP[] >=.06)
RWA_LOW[] = 3
RWA_MID[] = 7
RWA_HI[] = 0
Pay_Type2[] = 1
;Pay_Type1[] = 0
ELSE
RWA_MID[] = 0
Pay_Type2[] = 0
Endif

;Best
If (x[] and RWA_SXP[] >=.075)
RWA_LOW[] = 3
RWA_MID[] = 7
RWA_HI[] = 10
Pay_Type3[] = 1
;Pay_Type2[] = 0
;Pay_Type1[] = 0
ELSE
RWA_HI[] = 0
Pay_Type3[] = 0
Endif

;Tight Pay
If (RWA_SXP[] >=.06 and SXP[] <=.05 and  Vshl[] <=.4)
RWA_XPLOT[]=10
Pay_Type4[] = 1
Cum_Pay[] = 1
Else
RWA_XPLOT[]=0
Pay_Type4[] = 0
Endif

;Wet
;If (x[] and RWA_SXP[] <.045)
;Pay_Type5[] = 1
;Else
;Pay_Type5[] = 0
;Endif


;Tight or Wet
If (RWA_SXP[] <.045)
RWA_NEG[] = 3
Pay_Type5[] = 1
Else
RWA_NEG[] = 0
Pay_Type5[] = 0
Endif

;Shaley Interval
If (Vshl[] >=.4)
Hi_Vshl[] = 5
Else
Hi_Vshl[] = 0
```

```
Hi_Vshl[] = 0
Endif
Endif
Endif
Endif



;************************End of RWA**********************************************************************************

;             TOTAL CLAY CORRECTED Sw
b[] = ( Swb[] * ( 1- (Rw / Rwb ) ) / 2 )
SwT[] = b[] + sqrt( b[]^2 + (Rw / Rwa[]) )
SwT[] = min( 1, max( Vshl[], SwT[] ) )
bx[] = ( Swbx[] * ( 1- (Rw / Rwb ) ) / 2 )
SwTx[] = b[] + sqrt( b[]^2 + (Rw / Rwax[]) )
SwTx[] = min( 1, max( Vshl[], SwTx[] ) )

;               Calculate SW Total
;Y[] = Vshl[] * ( RwBnd - Rw ) / ( 2 * RwBnd )
;SwT[] = Y[] +( (Rw / ( RT[] * PHIA[]^2 ) ) + Y[]^2 )^.5
;SwT[] = min( 1, max( Vshl[], SwT[] ) )

;              EFFECTIVE WATER SATURATION
;If ( Vshl[] < .75 ) Then SwE[] = ( SwT[] - Vshl[] ) / ( 1 - Vshl[] ) else SwE[] = SwT[]
If ( Vshl[] < .75 ) Then SwE[] = ( SwT[] - Swb[] ) / ( 1 - Swb[] ) else SwE[] = SwT[]
If ( Vshl[] < .75 ) Then SwEx[] = ( SwTx[] - Swbx[] ) / ( 1 - Swbx[] ) else SwEx[] = SwTx[]

;              Water Bearing Resistivity
RoDW[] = (1 / (PHIT[]^m)) * (Rw * Rwb / (Rw * Vshl[] + Rwb * (1 - Vshl[])))
RoDWx[] = (1 / (PHITx[]^m)) * (Rw * Rwb / (Rw * Vshl[] + Rwb * (1 - Vshl[])))

;              Calculate SW Archie
;SwA[] = ( RoDW[] / RT[] )^0.5
SwAd[] = sqrt( a * Rw / (RT[] * PHID[] ^m) )
SwA[] = (a * Rw / ( RT[] * PHIA[]^m) )^(1/n)

;             Flushed Zone Water Saturation
Sxo[] = (a * <Rmf> / ( Res10[] * PHIE[]^m) )^(1/n)

;              Conductivity
If (IsNull (RT[])) Then COND[] = COND[] Else COND[] = 1000/RT[]

;              Bulk Volume Water
BVW[] = PHIE[] * SwE[]
BVWx[] = PHIEx[] * SwEx[]

;              Bulk Volume Bound Water
BVWb[] = PHIE[] * Swb[]
BVWbx[] = PHIEx[] * Swbx[]

;*********** Doug Johnson Relative Perm Comparison ***********
;             Effective Oil Saturation
SoE[] = 1 - SwE[]
;              Bulk Volume Hydorcarbon
BVhc[] = PHIE[] - BVW[]
;              Relative Perm to Water
Kwtr[] = 10^(( ln(BVW[] * (1-SoE[])) / 0.59) + 5)
;              Relative Perm to Hydrocarbon
If (BVhc[]<0.001)
Khc[] = 0
Else
Khc[] = 10^(( ln(BVhc[] * (1-SwE[])) / 0.59) + 5)
End If
;              Relative Perm Total
Ktot[] = Kwtr[] + Khc[]
;              Perm Flags
If (Kwtr[] / Khc[] < 10) Then PmFlag[] = 2 Else PmFlag[] = 0
If (Khc[] / Ktot[] > 0.5) Then KhcFlag[] = 3 Else KhcFlag[] = 0

;*************************************************************************

;              Quick Look Perm Quality From Microlog
PermQual[] = <Rm> * 10

;              Pay Flags
If (BHF[] < 1) Then PAY[] = PHIA[] >=  PhiCutoff and SwE[]  <= SwCutoff and Vshl[]  <= VshCutoff Else PAY[] = 0
;If (BHF[] < 1) Then PayE[] = PHIE[] >=  PhiCutoff and SwE[]  <= SwCutoff and Vshl[]  <= VshCutoff Else PAY[] = 0
;NetPay[] = PAY[]*0.5
;NetPayE[] = PayE[]*0.5

If (IsNull (RHOB[])==1) Then NetPhi[]=(PHIS[] >= PhiCutoff and Vshl[] <= VshCutoff)*0.5 Else NetPhi[] = (PHID[] >= PhiCutoff and Vshl[] <= VshCutoff)*0.5
If (Vshl[] <= VshCutoff) Then NetPhiS[] = (PHIS[] >= PhiCutoff)*0.5 Else NetPhiS[] = 0

;             Induction from COND_IN
;If (IsNull (LLD[])==1) Then LLD[] = 1000 / COND_IN[] Else LLD[] = LLD[]
;             Hydrocarbon Pore Volume
;SoPhiH[] = (1-SwE[]) * PHIA[] * <Step>
SoPhiH[] = BVW[] - PHIE[]

;  Coates Permeability
K[] = (70 * PHIE[]^2 * (1- BVW[])/BVW[])^2

;              Rxo/Rt
RxoRt[] = ResS[] / RT[]

;              Bit Size Curve
Bit[] = <BitSize>

;                Rw Estimate from Water Sample
;Rw[] = RwSample * (TempSample + 6.77) / (TempFormation + 6.77)
Rw[] = 0.6 * (70 + 6.77) / (<temp> + 6.77)

;               GR_MWD for GeoSteering
If (IsNull(GR_MWD[]))
GR_MWD[] = GR[]
Else
```

```
Else
GR_MWD[] = GR_MWD[]
End If

;               Mineral Lithology Calculations
;U[] = PEF[] * ((RHOB[] + 0.1883) / 1.07)
;UMA[] = (U[] - 0.398 * PHIA[]) / (1 - PHIA[])
;DGA[] = (RHOB[] - PHIA[]) / (1 - PHIA[])
;               Mineral Proportions w/o PEF
;   Limestone
;Lime[] =(BA_PorLitho2420_10_JWA_Lime (PHINc[],RHOB[]) ) * (1 - Vshl[])
;Sand[]=(BA_PorLitho2420_10_JWA_Sand (PHINc[],RHOB[]) ) * (1 - Vshl[])
;Dolo[]=(BA_PorLitho2420_10_JWA_Dolo (PHINc[],RHOB[]) ) * (1 - Vshl[])

;               Missing RHOB
RHOB[] = -1*( (PHID[] * (RhoM-RhoF)) - RhoM)

;------------- Jay B. Chapman Lithology ----------------
;------------------POROSITY CURVES----------------
;***Cross Plot Porosity
XP_PHI[] = sqrt((PHID[] ^ 2 + PHIN[] ^ 2) / 2)
If (IsNull(XPLOT[])) Then XPLOT[] = XP_PHI[]
XPLOT[] = Max(0, XPLOT[])

;****shale corrected Neutron
PHINsc[]=PHIN[]-(Vshl[]*PhiNshl)

;-------------Lithology ----------------
; U=PE absorption coefficient
;DGA=apparent grain density, UMA=apparent matrix PE

;these are the equations used in PRIZM, not used here
;U = PEF * ((RHOB + 0.1883) / 1.07)
;UMA = (U - 0.398 * PHIA) / (1 - PHIA)
;not sure where 0.1883 or 1.07 numbers come from
;0.398 is for fresh water, 1.36 is for salt water

;Densities used below
;calcite
d1=2.71
;sand
d2=2.65
;dolomite
d3=2.85

;photoelectric absoprtion coefficient, specific to rock type
;calcite
u1=13.77
;sand
u2=4.88
;dolomite
u3=9.0

;UGA2 and DGA2 track very closely with PRIZM output in 4min model, good to use
UMA1[] = PEF[]*RHOB[]-XPLOT[]*0.398
DGA2[] = (RHOB[]-XPLOT[])/(1 - XPLOT[])
UMA2[] = UMA1[]/(1-XPLOT[])

;Density Porosity using DGA (Jack Austin)
PHID_DGA[] = (DGA[] - RHOB[]) / (DGA[] - RhoF)

;***equations used by JBC below, do not turn on, do not include shale until later, 3 row matrix
;P1 + P2  + P3 = 1
;d1P1 + d2P2 + d3P3 = DGA
;u1P1 + u2P2 + u3P3 = UMA

;documentation of matrix operations follow
;multiply row 1 by (-u1) and add to row 3, action performed on above matrix
;P1 + P2  + P3 = 1
;d1P1 + d2P2 + d3P3 = DGA
;P2(u2-u1)  + P3(u3-u1) = UMA-u1

;multiply row 1 by (-d1) and add to row 2, action performed on above matrix, still ignoring shale
;P1 + P2  + P3 = 1
;P2(d2-d1)  + P3(d3-d1) = DGA-d1
;P2(u2-u1)  + P3(u3-u1) = UMA-u1

;multiply row 2 by (u2-u1) and mutliple row 3 by (d2-d1)
;P1 + P2  + P3 = 1
;P2(d2-d1)(u2-u1)  + P3(d3-d1)(u2-u1) = (DGA-d1)(u2-u1)
;P2(u2-u1)(d2-d1)  + P3(u3-u1)(d2-d1) = (UMA-u1)(d2-d1)

;multiply row 2 by -1 and add to row 3
;P1 + P2  + P3 = 1
;P2(d2-d1)(u2-u1)  + P3(d3-d1)(u2-u1) = (DGA-d1)(u2-u1)
;[P3(u3-u1)(d2-d1)]-[P3(d3-d1)(u2-u1)] = [(UMA-u1)(d2-d1)]-[(DGA-d1)(u2-u1)]

;divide row 2 by (u2-u1), performed on above matrix, now have isolated P3 in row 3
;P1 + P2  + P3 = 1
;P2(d2-d1) + P3(d3-d1) = (DGA-d1)
;[P3(u3-u1)(d2-d1)]-[P3(d3-d1)(u2-u1)] = [(UMA-u1)(d2-d1)]-[(DGA-d1)(u2-u1)]

;subtituting coefficients in for parameters in row 3 above, solving for P3
;[P3a]-[P3b] = [(UMA-u1)(c)]-[(DGA-d1)(d)]
;a=(u3-u1)(d2-d1)
;b=(d3-d1)(u2-u1)
;c=(d2-d1)
;d=(u2-u1)

coeffa= (u3-u1)*(d2-d1)
coeffb=(d3-d1)*(u2-u1)
coeffc=(d2-d1)
coeffd=(u2-u1)

;more substitutions, solving in parts
;P3(a-b)=cUMA-u1c -dDGA -d1d
;rearranging
;P3=[cUMA-dDGA-(u1c-d1d)]/(a-b)
```

```
;P3=[cUMA-dDGA-(u1c-d1d)]/(a-b)
;P3=[cUMA-dDGA-e]/f

;more coefficients
;e=(u1c-d1d)
;f=(a-b)
coeffe=(u1*coeffc)-(d1*coeffd)
coefff=coeffa-coeffb

;finally solving for P3 (dolomite)
exP3[]=((coeffc*UMA2[])-(coeffd*DGA2[])-(coeffe))/coefff

;plugging P3 back in to solve for P2
;coefficient c = (d2-d1) from above
;P2(d2-d1) + P3(d3-d1) = (DGA-d1)
;P2c = (DGA-d1)-P3(d3-d1)
;P2 = [(DGA-d1)-P3(d3-d1)]/c

;more coeffecients
;g=d3-d1
coeffg=(d3-d1)

;P2=sand
exP2[]=((DGA2[]-d1)-(exP3[]*coeffg))/coeffc

;P1=calcite
;now solve for P1, these values still do not include shale
exP1[]=1-exP2[]-exP3[]

;relative volume of mineral, do not include shale
MinrlP1[]=max(0,exP1[])/(max(0,exP1[])+max(0,exP2[])+max(0,exP3[]))
MinrlP2[]=max(0,exP2[])/(max(0,exP1[])+max(0,exP2[])+max(0,exP3[]))
MinrlP3[]=1-MinrlP1[]-MinrlP2[]

;-------------------Lithology using only Neutron and Density Curves------------------

;***equations used by JBC below, do not turn on, do not include shale until later, 3 row matrix
;P1 + P2  + P3 = 1
;d1P1 + d2P2 + d3P3 = DGA
;x1P1 + x2P2 + x3P3 = PhiXc

;x=coeffiecent for amount of cross over in dens-neut (shale corrected)
;limestone (x1)
x1=0
;these are user entered parameters. dolo is probably pretty good
;not sure what good sand value is, .1 to .15 probably reasonably range
;sand x2 =.125
;dolo x3 = -.075

;***Calculate crossover using shale corrected Neutron
;can view these curves to get a sense of what the range of cross-over is and calibrate x2, x3 values above if lith known
PhiX[]=PHID[]-PHIN[]
PhiXc[]=PHID[]-PHINsc[]

;documentation of matrix operations follow
;tried dropping P1 term out in row 3 since x1=0, but equation didn't work
;multiply row 1 by (-d1) and add to row 2
;P1 + P2  + P3 = 1
;P2(d2-d1)  + P3(d3-d1) = DGA-d1
;x1P1 + x2P2 + x3P3 = PhiXc

;multiply row 1 by (-x1) and add to row 3
;P1 + P2  + P3 = 1
;P2(d2-d1)  + P3(d3-d1) = DGA-d1
;P2(x2-x1)  + P3(x3-x1) = PhiXc-x1

;multiply row 2 by (x2-x1) and mutliply row 3 by (d2-d1)
;P1 + P2  + P3 = 1
;P2(d2-d1)(x2-x1)  + P3(d3-d1)(x2-x1) = (DGA-d1)(x2-x1)
;P2(x2-x1)(d2-d1) + P3(x3-x1)(d2-d1) = (PhiXc-x1)(d2-d1)

;multiply row 2 by -1 and add to row 3
;P1 + P2  + P3 = 1
;P2(d2-d1)(x2-x1)  + P3(d3-d1)(x2-x1) = (DGA-d1)(x2-x1)
;[P3(x3-x1)(d2-d1)]-[P3(d3-d1)(x2-x1)] = [(PhiXc-x1)(d2-d1)]-[(DGA-d1)(x2-x1)]

;divide row 2 by (x2-x1), performed on above matrix, now have isolated P3 in row 3
;P1 + P2  + P3 = 1
;P2(d2-d1) + P3(d3-d1) = (DGA-d1)
;[P3(x3-x1)(d2-d1)]-[P3(d3-d1)(x2-x1)] = [(PhiXc-x1)(d2-d1)]-[(DGA-d1)(x2-x1)]

;subtituting coefficients in for parameters in row 3 above, solving for P3
;[P3ax]-[P3bx] = [(PhiXc-x1)(cx)]-[(DGA-d1)(dx)]
;ax=(x3-x1)(d2-d1)
;bx=(d3-d1)(x2-x1)
;c=(d2-d1), from above
;dx=(x2-x1)

coeffax= (x3-x1)*(d2-d1)
coeffbx=(d3-d1)*(x2-x1)
coeffdx=(x2-x1)

;more substitutions, solving in parts
;P3(ax-bx)=c*PhiXc-u1c -dxDGA -d1dx
;rearranging
;P3=[cPhiXc-dxDGA-(x1c-d1d)]/(a-b)
;P3=[cPhiXc-dxDGA-ex]/fx

;more coefficients
;ex=(x1c-d1dx)
;fx=(ax-bx)
coeffex=(x1*coeffc)-(d1*coeffdx)
coefffx=coeffax-coeffbx

;finally solving for P3 (dolomite)
xxP3[]=((coeffc*PhiXc[])-(coeffdx*DGA2[])-(coeffex))/coefffx
```

```
;plugging P3 back in to solve for P2
;coefficient c = (d2-d1) from above
;P2(d2-d1) + P3(d3-d1) = (DGA-d1)
;P2c = (DGA-d1)-P3(d3-d1)
;P2 = [(DGA-d1)-P3(d3-d1)]/c

;more coeffecients
;g=d3-d1, from above

;P2=sand
xxP2[]=((DGA2[]-d1)-(xxP3[]*coeffg))/coeffc

;P1=calcite
;now solve for P1, these values still do not include shale
xxP1[]=1-xxP2[]-xxP3[]

;relative volume of mineral, do not include shale
xMinrlP1[]=max(0,xxP1[])/(max(0,xxP1[])+max(0,xxP2[])+max(0,xxP3[]))
xMinrlP2[]=max(0,xxP2[])/(max(0,xxP1[])+max(0,xxP2[])+max(0,xxP3[]))
xMinrlP3[]=1-xMinrlP1[]-xMinrlP2[]

;--------------------ADD SHALE VOLUME AND DISPLAY-------------------------

;relative volumes with shale added
VP1[]=(1-Vshl[])*MinrlP1[]
VP2[]=(1-Vshl[])*MinrlP2[]
VP3[]=(1-Vshl[])*MinrlP3[]

If (IsNull(PEF[])) Then VP1[]=(1-Vshl[])*xMinrlP1[]
If (IsNull(PEF[])) Then VP2[]=(1-Vshl[])*xMinrlP2[]
If (IsNull(PEF[])) Then VP3[]=(1-Vshl[])*xMinrlP3[]

;***for testing only to compare ND model with PE model, do not use
xVP1[]=(1-Vshl[])*xMinrlP1[]
xVP2[]=(1-Vshl[])*xMinrlP2[]
xVP3[]=(1-Vshl[])*xMinrlP3[]
;for test display, do not use
xxP1dsp[]=xVP1[]
xxP2dsp[]=xVP1[]+xVP2[]
xxP3dsp[]=xVP1[]+xVP2[]+xVP3[]
xxP4dsp[]=1

;for display only
exP1dsp[]=VP1[]
exP2dsp[]=VP1[]+VP2[]
exP3dsp[]=VP1[]+VP2[]+VP3[]
exP4dsp[]=1
; Replace Prizm 3mineral display curves - JWA
P1dsp[]=VP1[]
P2dsp[]=VP1[]+VP2[]
P3dsp[]=VP1[]+VP2[]+VP3[]

;Mineral sums for Carbonate vs Silica template - JWA
TotCarb[] = MinrlP1[] + MinrlP3[]
If (IsNull(TotCarb[]))
TotCarb[] = xMinrlP1[] + xMinrlP3[]
Else
TotCarb[] = TotCarb[]

;Clean Gross
If ( Vshl[] < .5 ) Then GrossCln[] = <Step>

;gross pay
If (GrossCln[] = <Step>) Then GrossPay[] = 1
```