

1/14

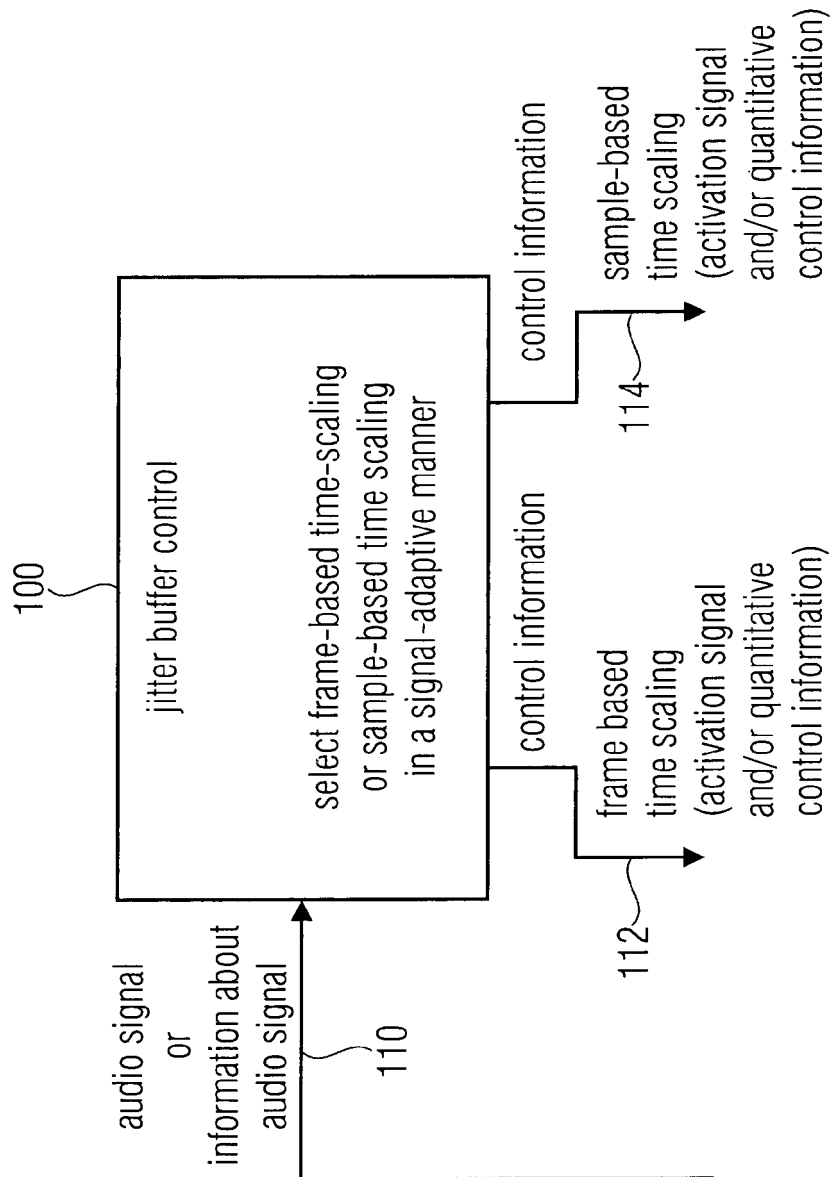


FIG 1

2/14

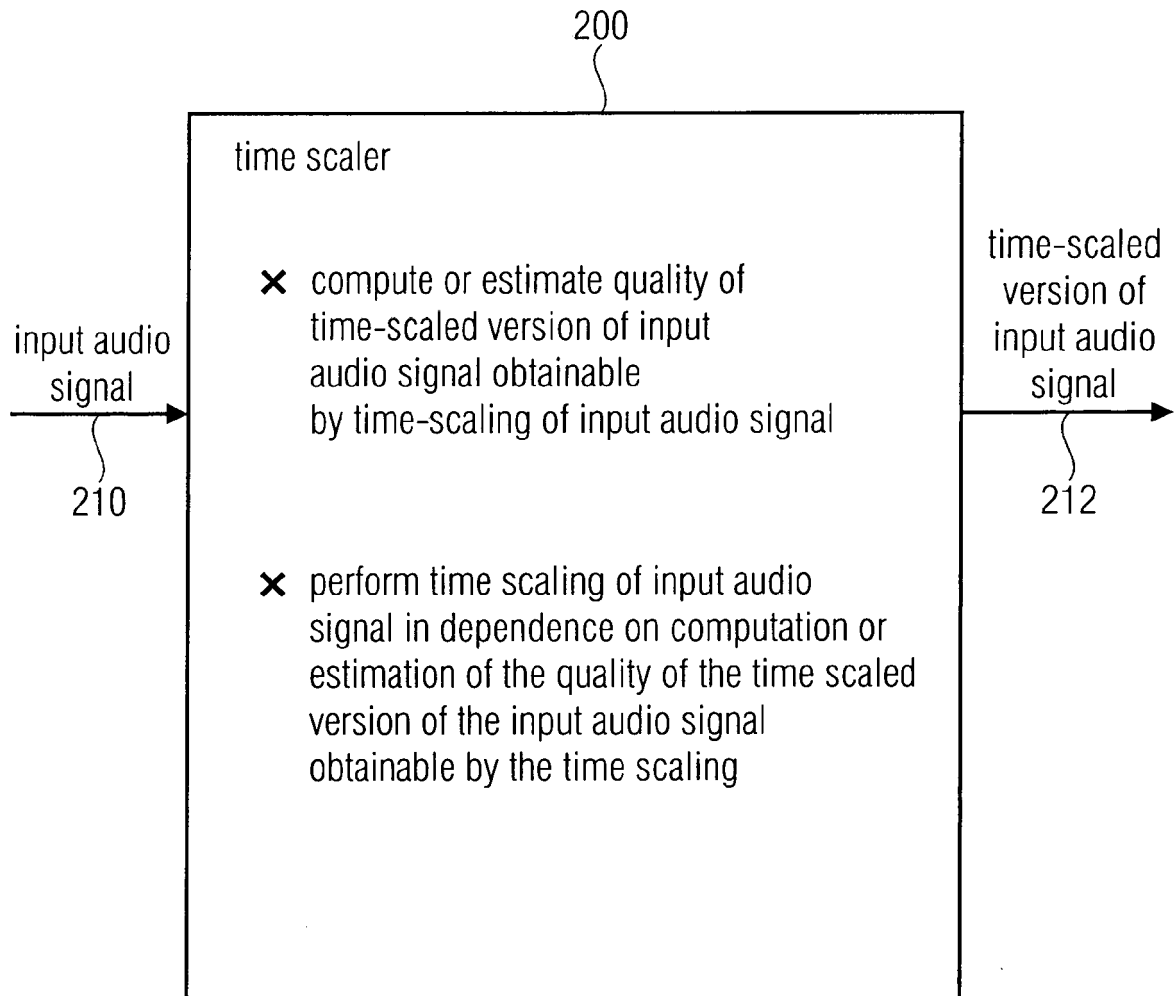


FIG 2

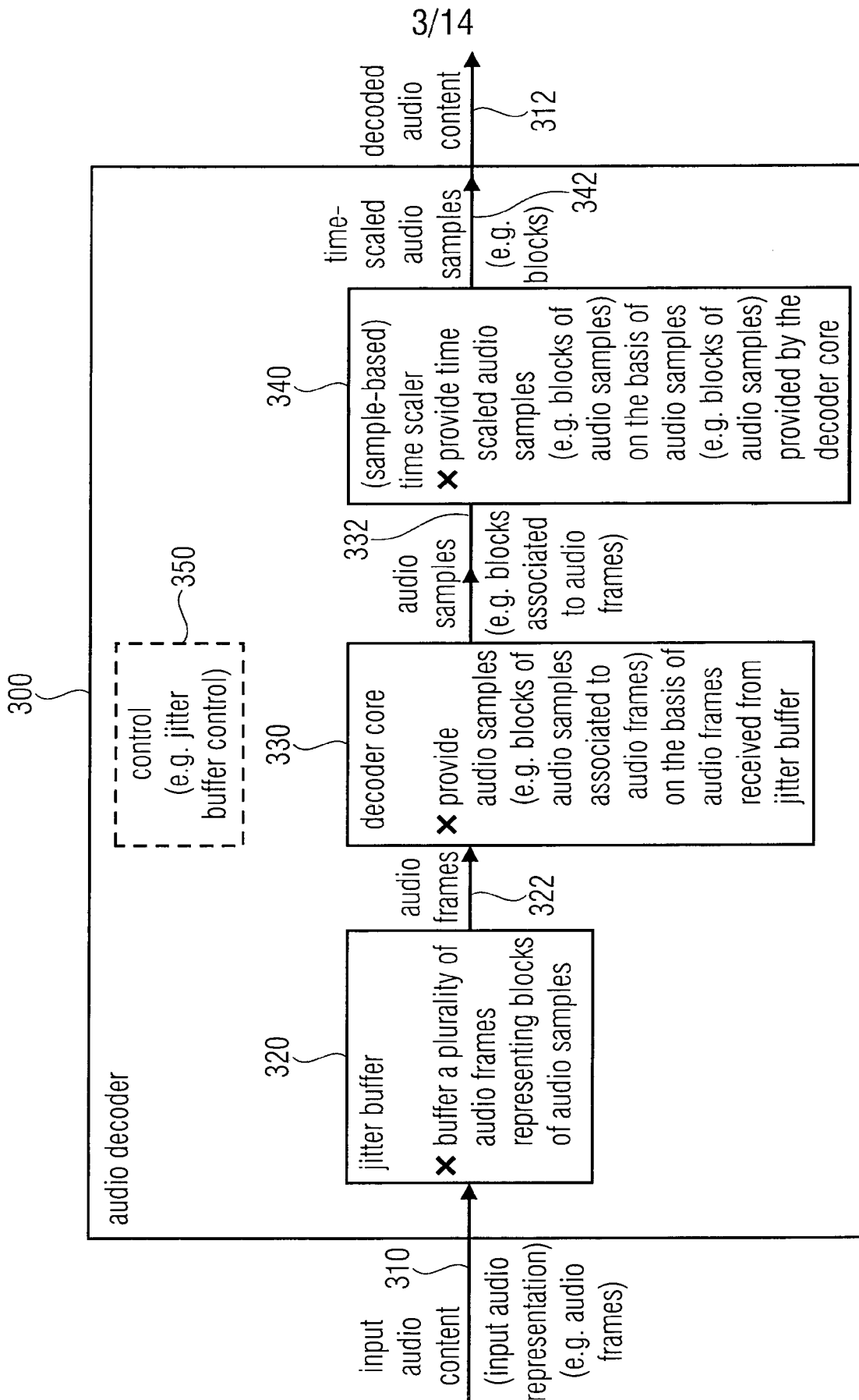


FIG 3

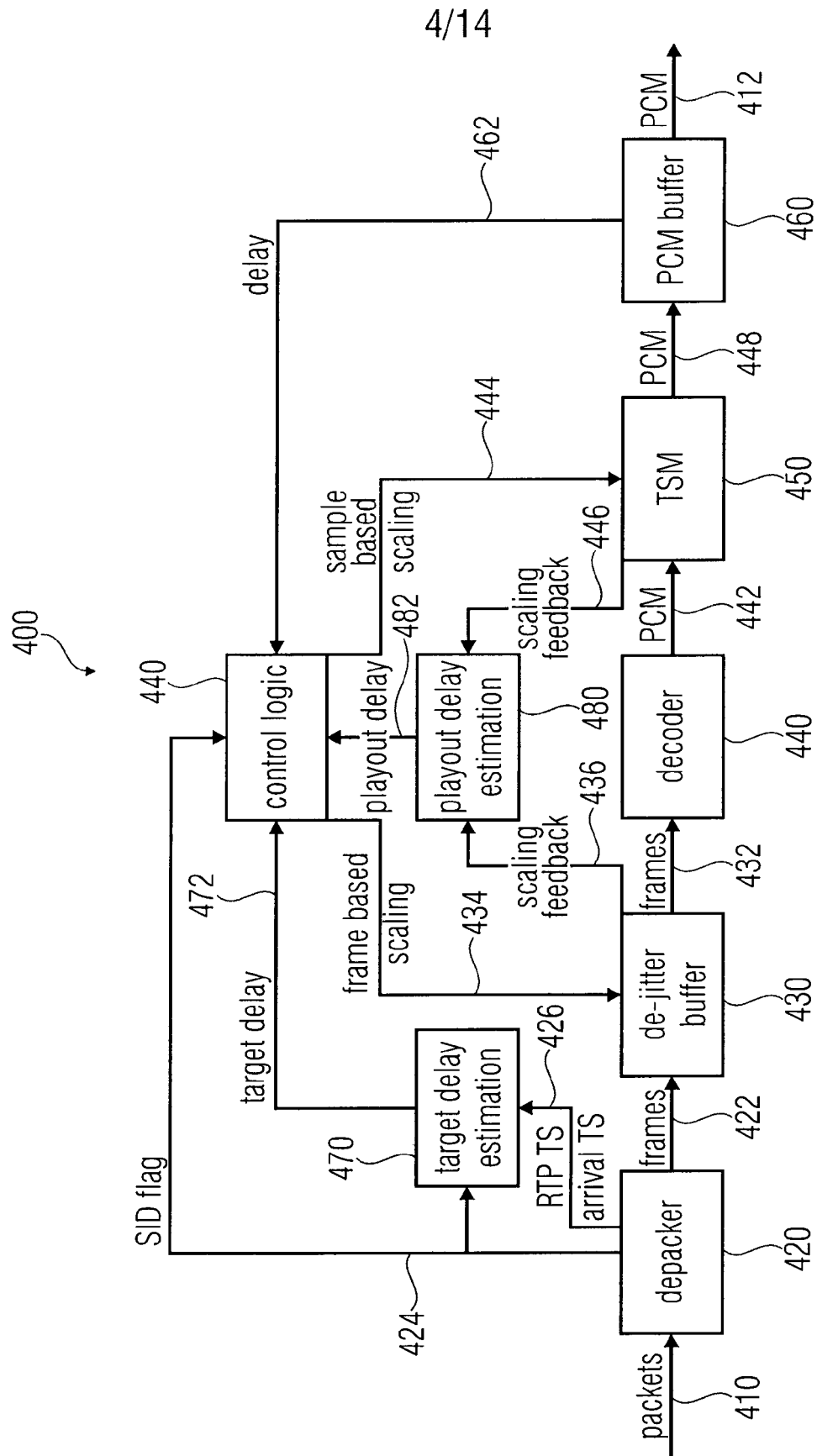


FIG 4

5/14

```

soundCardFrameSize = sampleRate * 20 / 1000;
while(pcmBuffer_nReadableSamples() < soundCardFrameSize) {
    accessUnit = deJitterBuffer_pop();           ← 510
    audioSamples = decoder_decodeOrConceal(accessUnit); ← 512
    tsm_scale(audioSamples);                     ← 514
    pcmBuffer_write(audioSamples);              ← 516
}
audioSamples = pcmBuffer_read(soundCardFrameSize); ← 520
soundcard_push(audioSamples);                  ← 522

```

FIG 5

```

rtpTimeDiff = rtpTimeStamp - prevRtpTimeStamp; ← 610
rcvTimeDiff = rcvTime - prevRcvTime;           ← 612
/* convert time base to milliseconds */
rtpTimeTicks = rtpTimeStamp * sysTimeScale / rtpTimeScale; ← 614
rtpTimeDiff = rtpTimeDiff * sysTimeScale / rtpTimeScale; ← 616
delay = rcvTimeDiff - rtpTimeDiff + prevDelay; ← 618
offset = rcvTime - rtpTimeTicks;                ← 620
longTermFifo_add( delay, offset, rtpTimeTicks ); ← 622
/* backup values of current packet */
prevRtpTimeStamp = rtpTimeStamp;                }
prevRcvTime      = rcvTime;                     ← 624
prevDelay        = delay;

```

FIG 6

```

targetMax      = shortTermJitter + 60;           ← 710
targetMin      = min( longTermJitter + 20, targetMax ); ← 712
targetDtx      = min( longTermJitter, shortTermJitter ); ← 714
targetStartUp  = ( targetMin + targetMax ) / 2;   ← 716

```

FIG 7

6/14

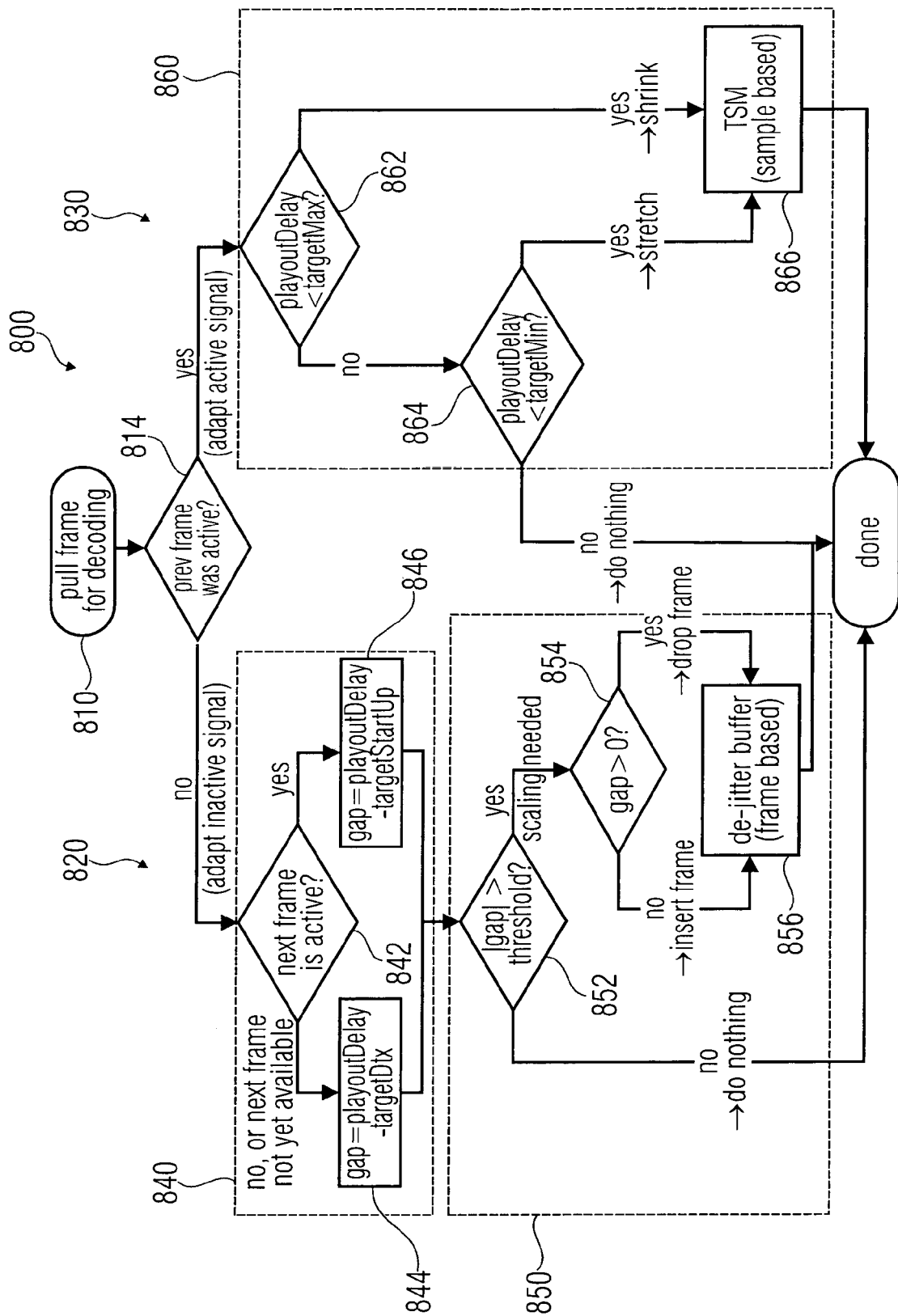


FIG 8

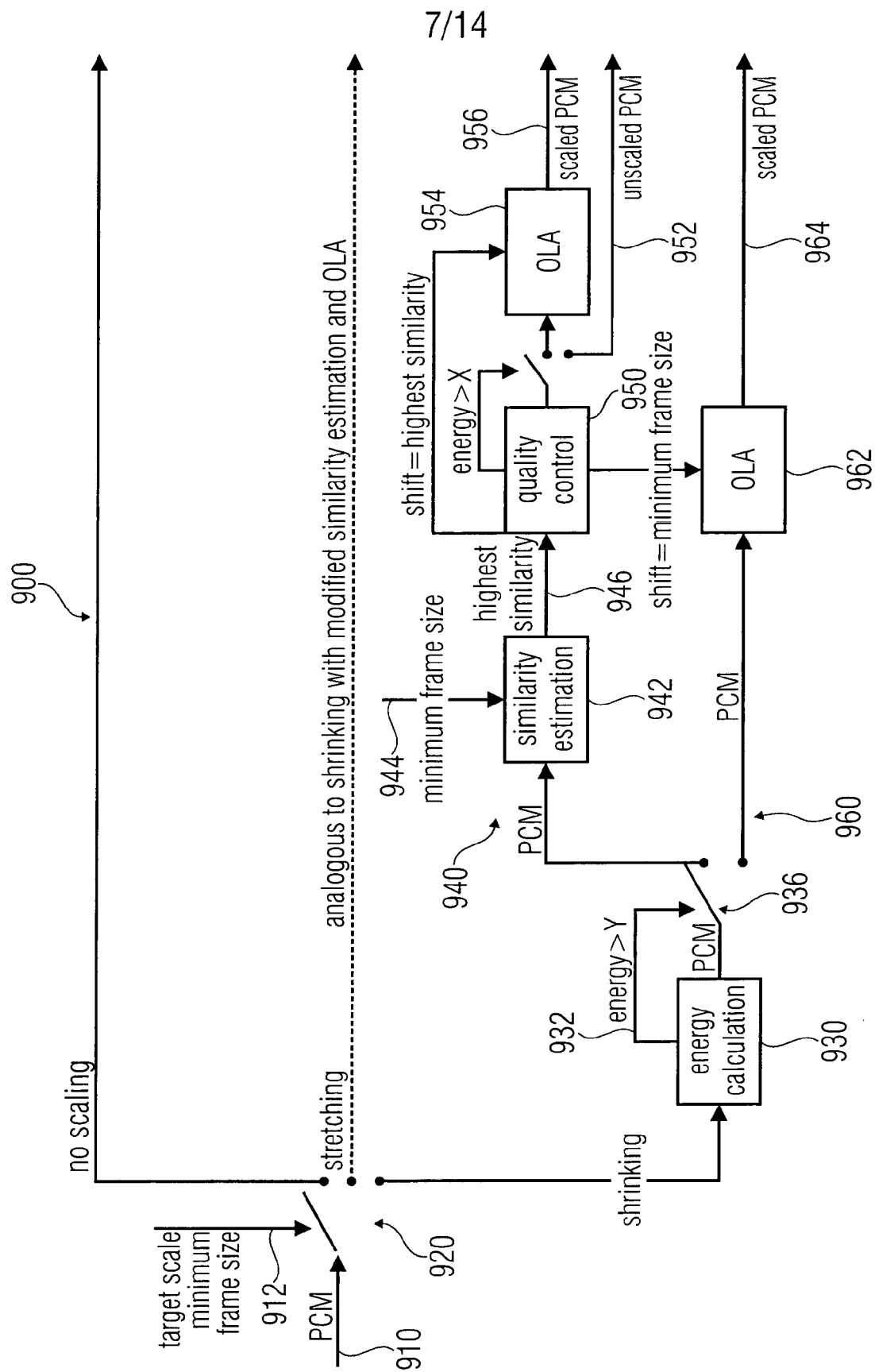


FIG 9

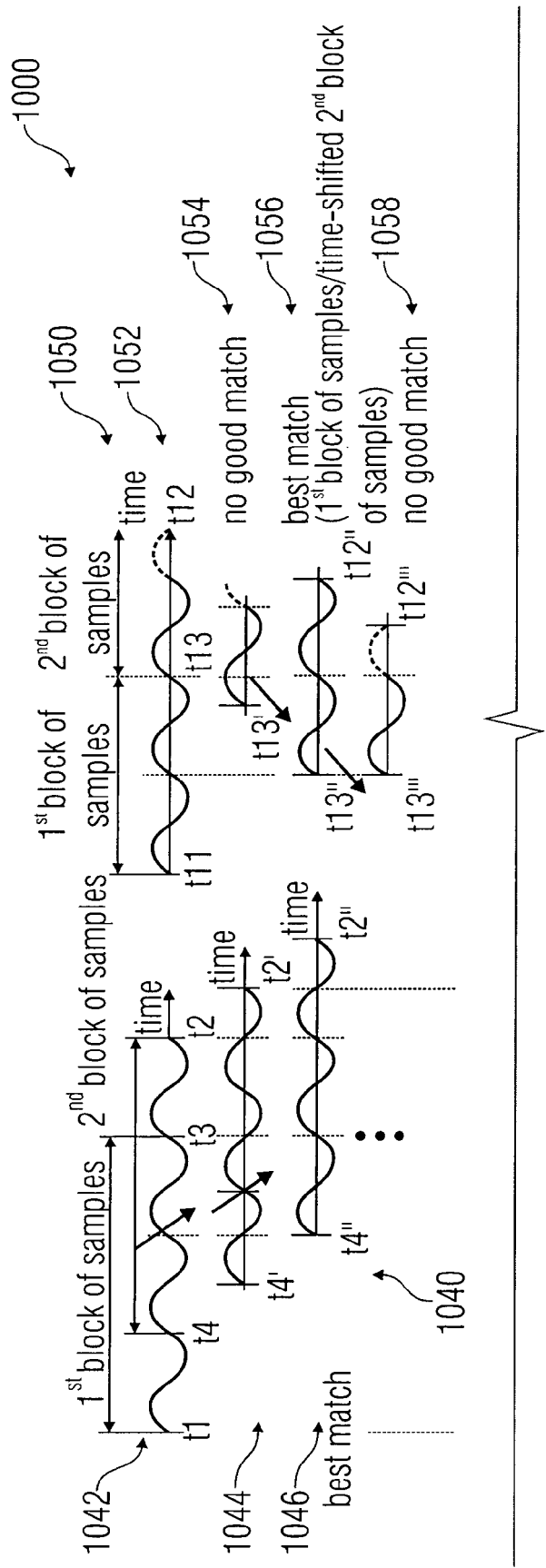
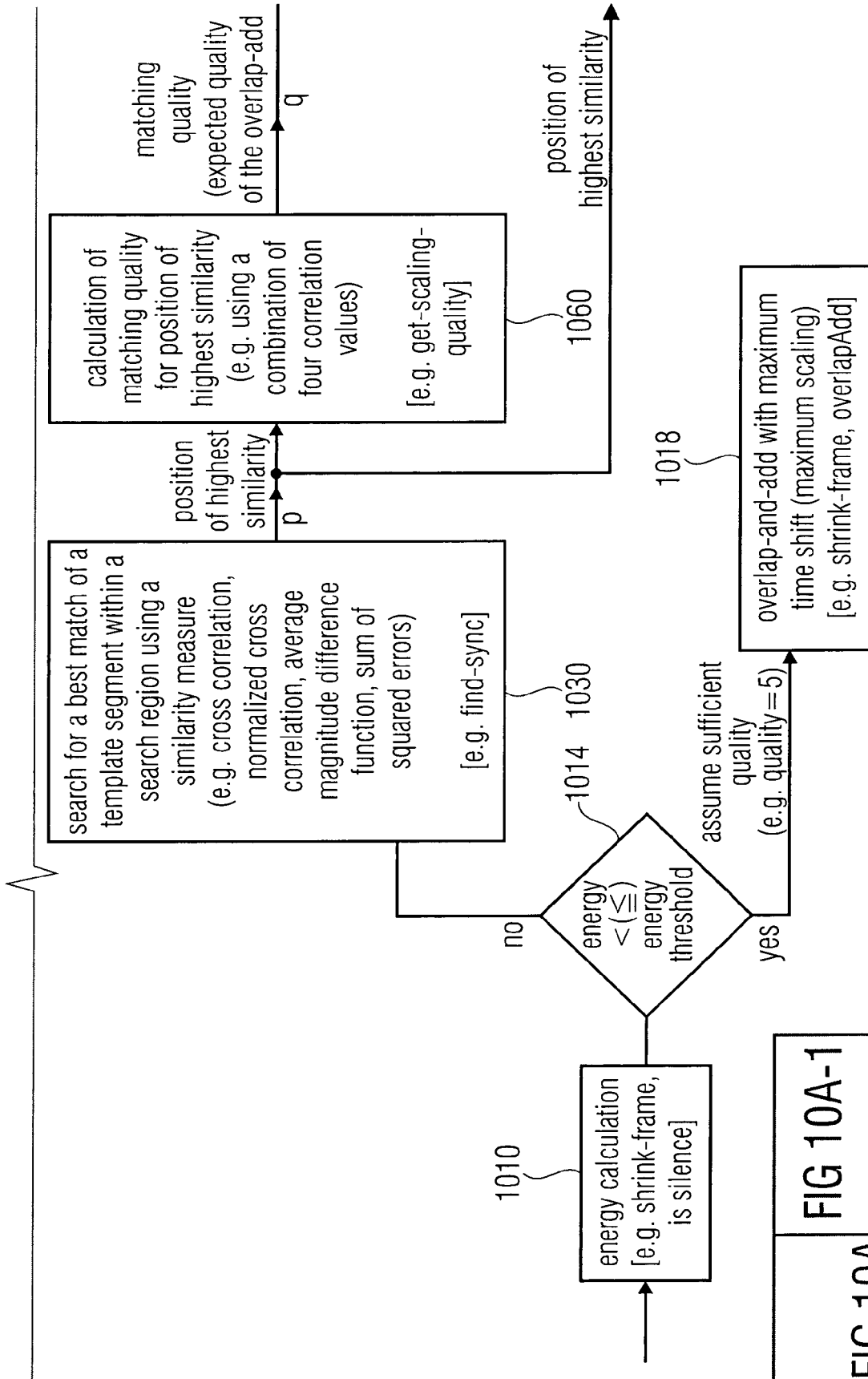


FIG 10A-1

FIG 10A	FIG 10A-1
	FIG 10A-2

9/14



10/14

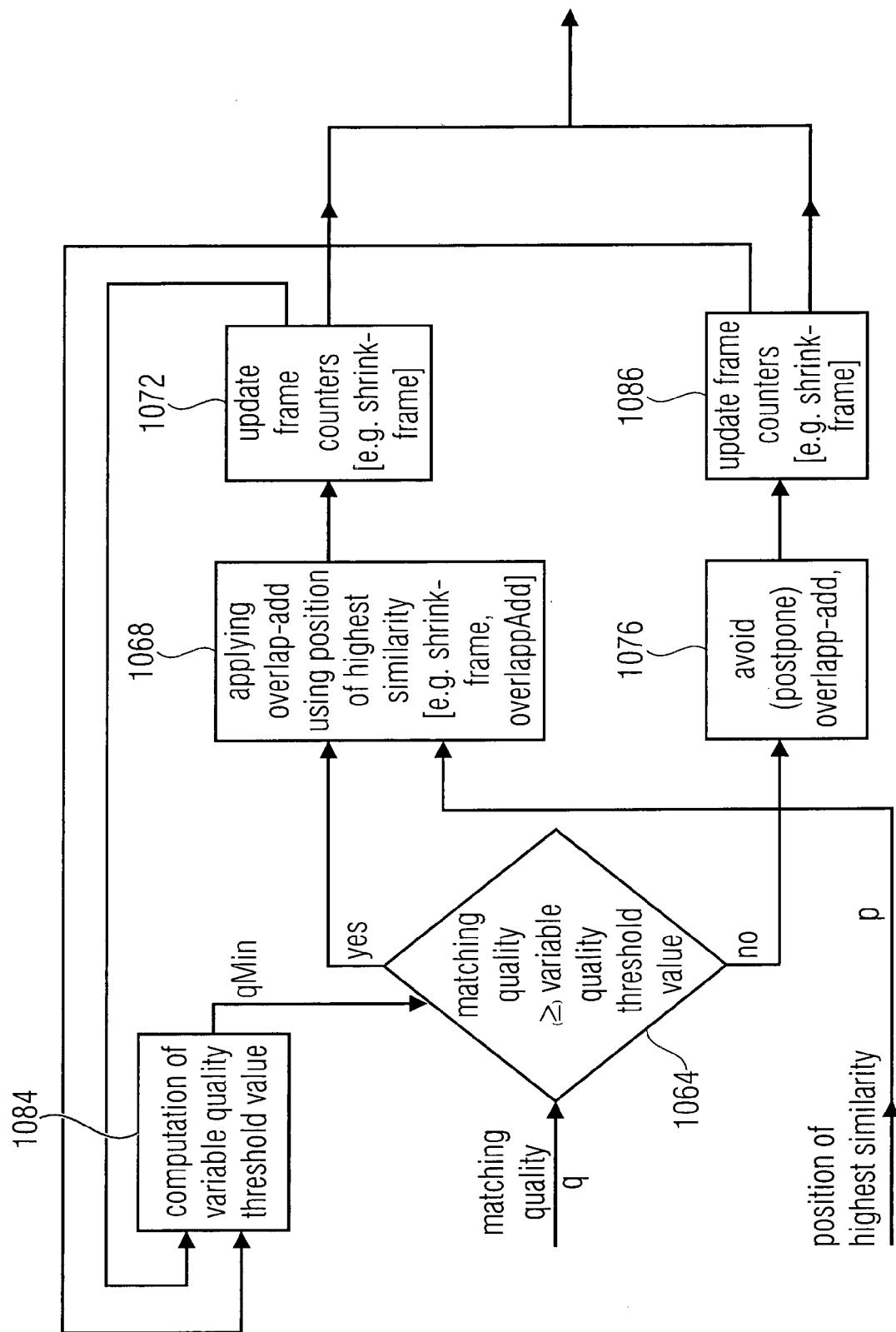


FIG 10B

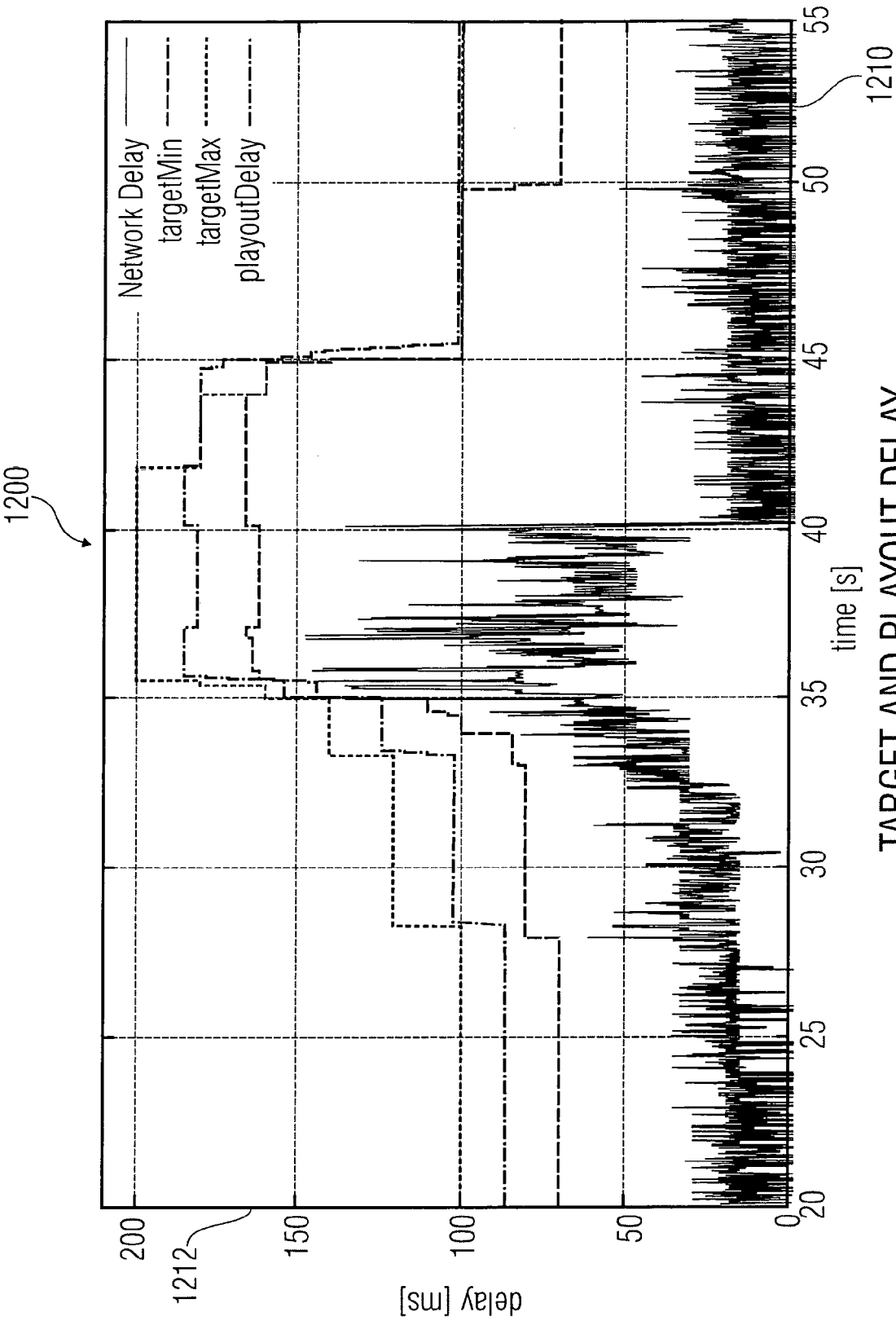
11/14

```

qMinInitial  = 1;                               ← 1110
qualityRise   = 4;                               ← 1112
qualityRed    = 4;                               ← 1114
p = similarityEstimation();                       ← 1116
q = c(p) * c(2*p) + c(3/2*p) * c(1/2*p);
qMin = qMinInitial - (nNotScaled * 0.1) + (nScaled * 0.2); ← 1118
if(q >= qMin) { /* sufficient quality, apply time-scaling */ ← 1120
    applyOverlapAdd();                             ← 1122
    if( nNotScaled > 0 )                            } ← 1124
        nNotScaled = nNotScaled - 1;
    if( nScaled < qualityRise )                      } ← 1126
        nScaled = nScaled + 1;
}
else { /* not sufficient, postpone time-scaling */
    if( nNotScaled < qualityRed )                    } ← 1128
        nNotScaled = nNotScaled + 1;
    if( nScaled > 0 )                                } ← 1130
        nScaled = nScaled - 1;
}

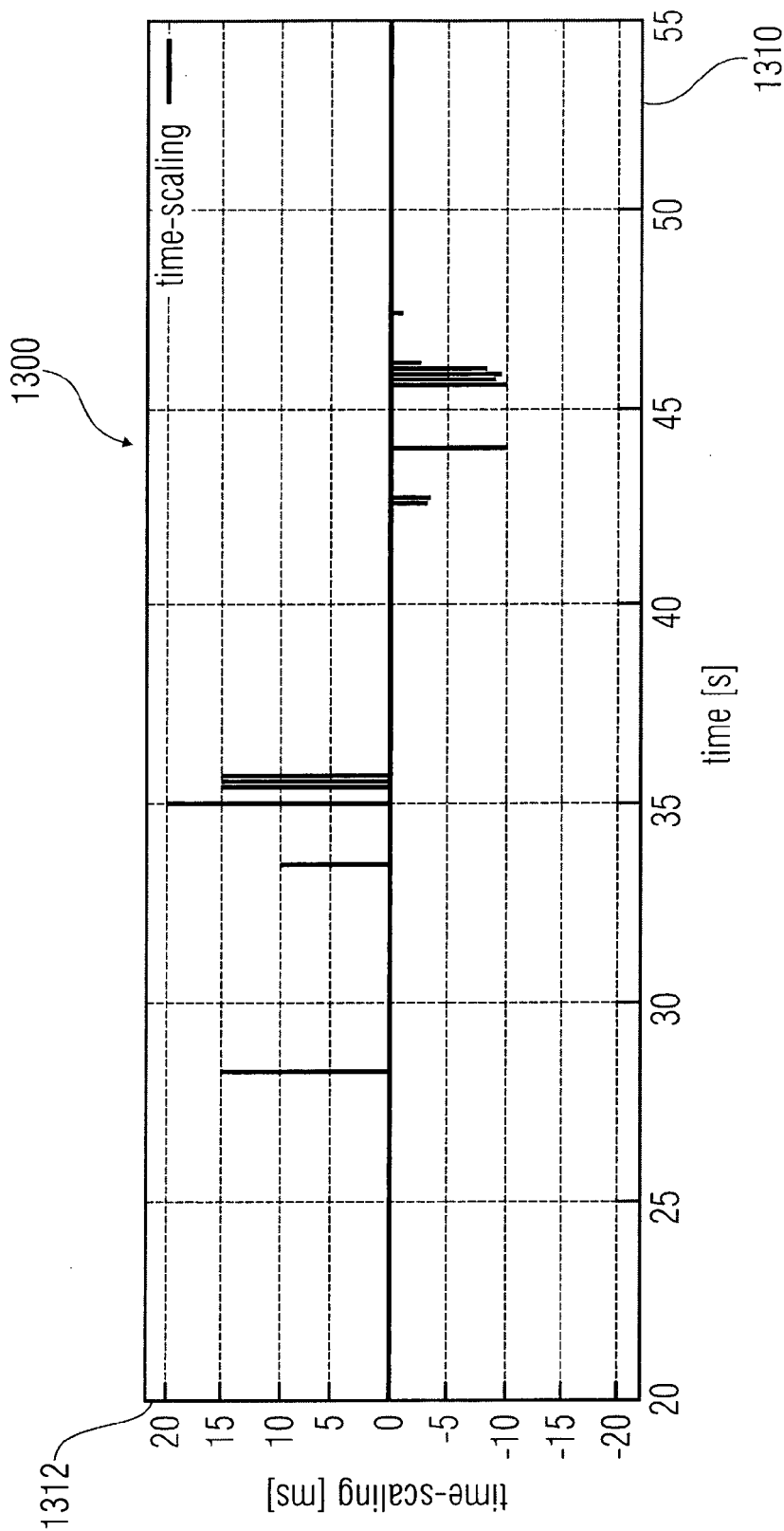
```

FIG 11



TARGET AND PLAYOUT DELAY

FIG 12



TIME-SCALING
FIG 13

14/14

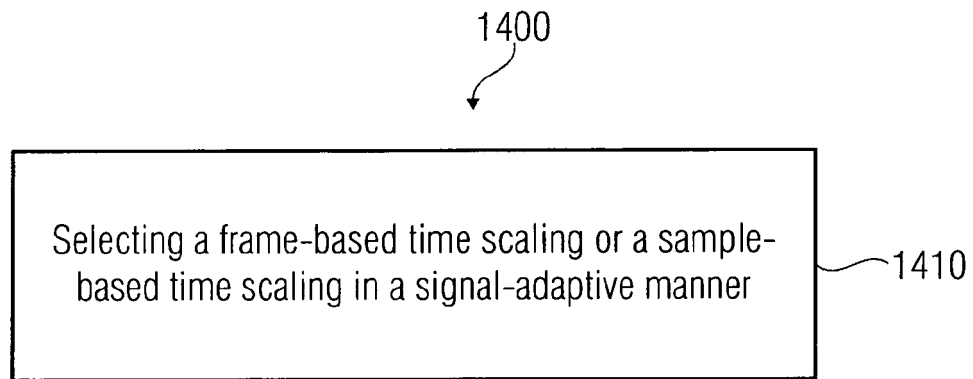


FIG 14

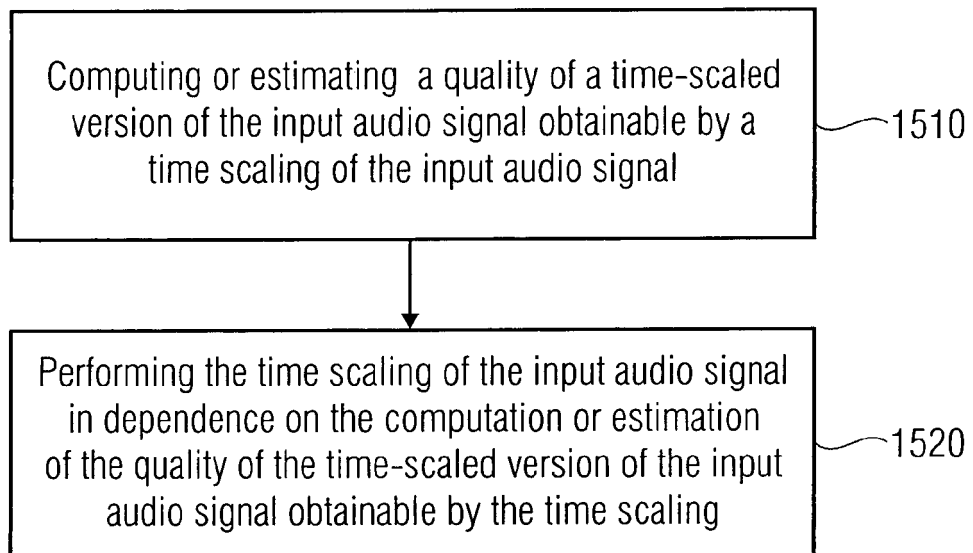


FIG 15