

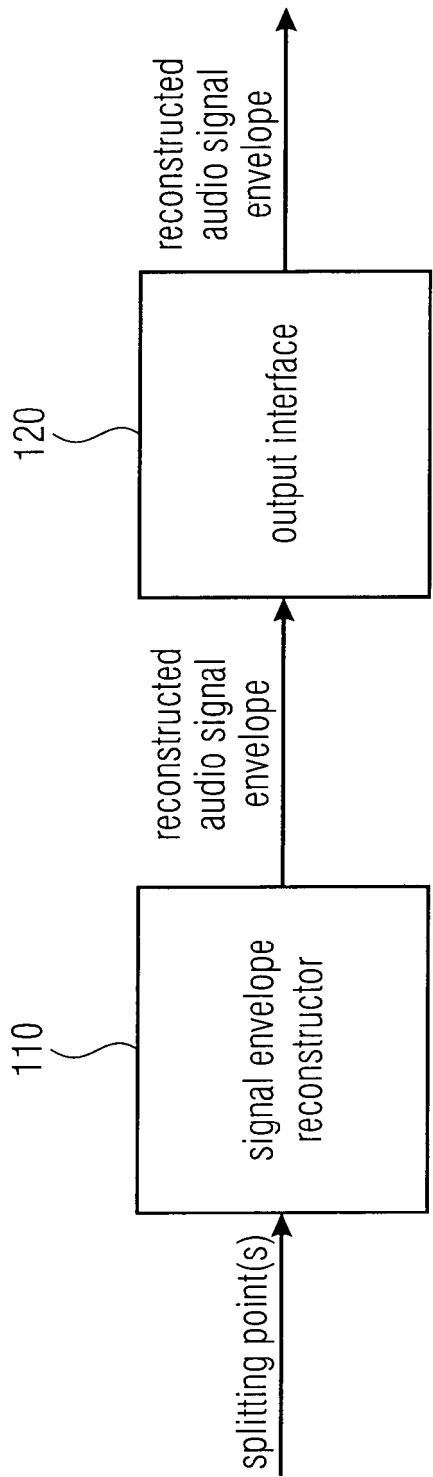


FIG 1

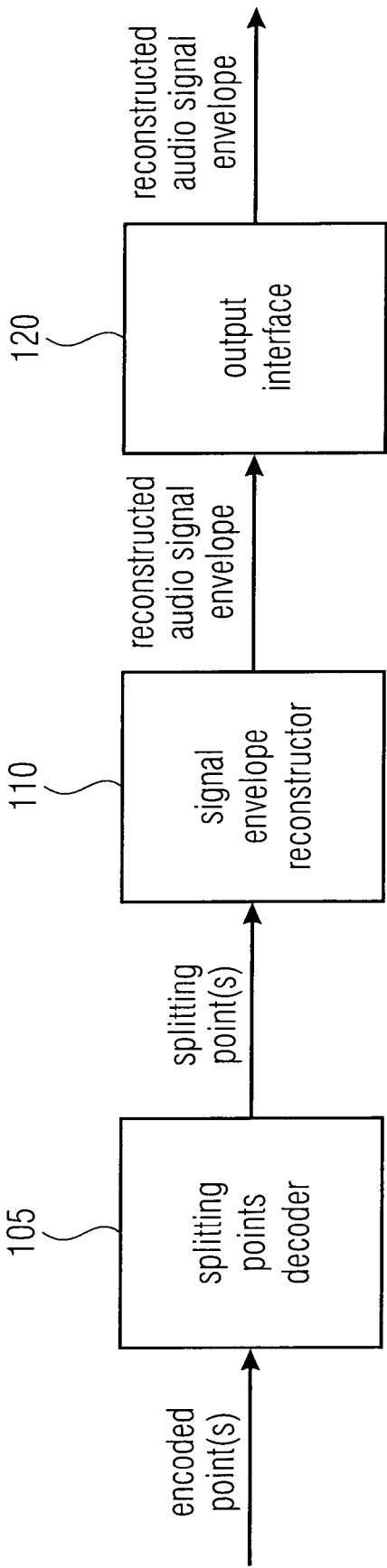


FIG 2

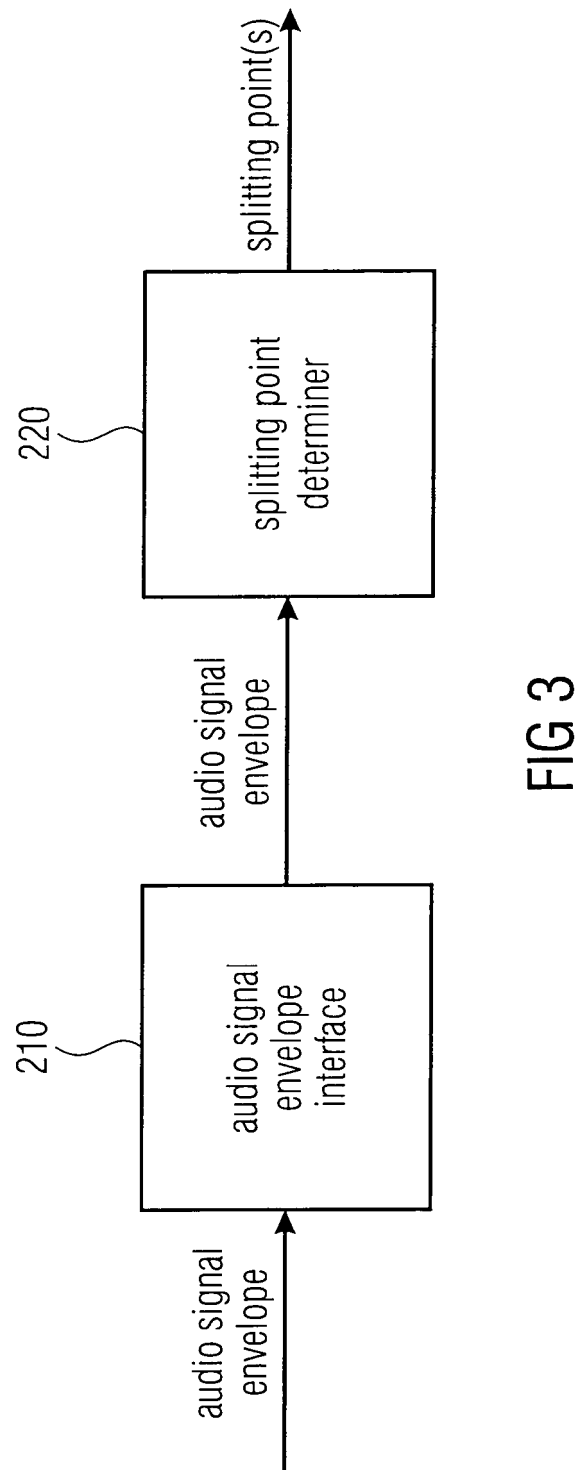


FIG 3

4/17

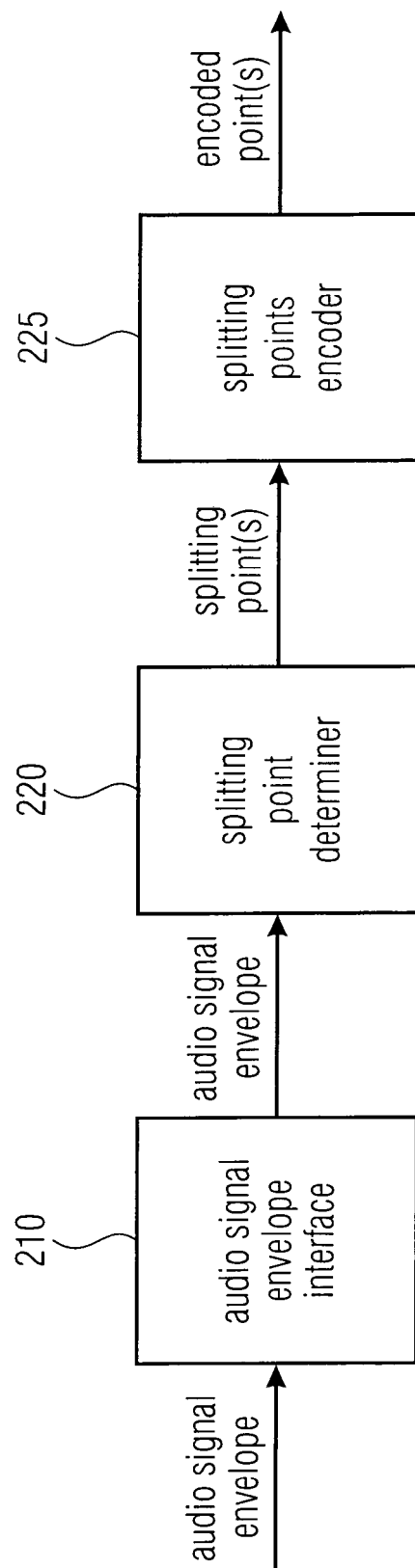


FIG 4

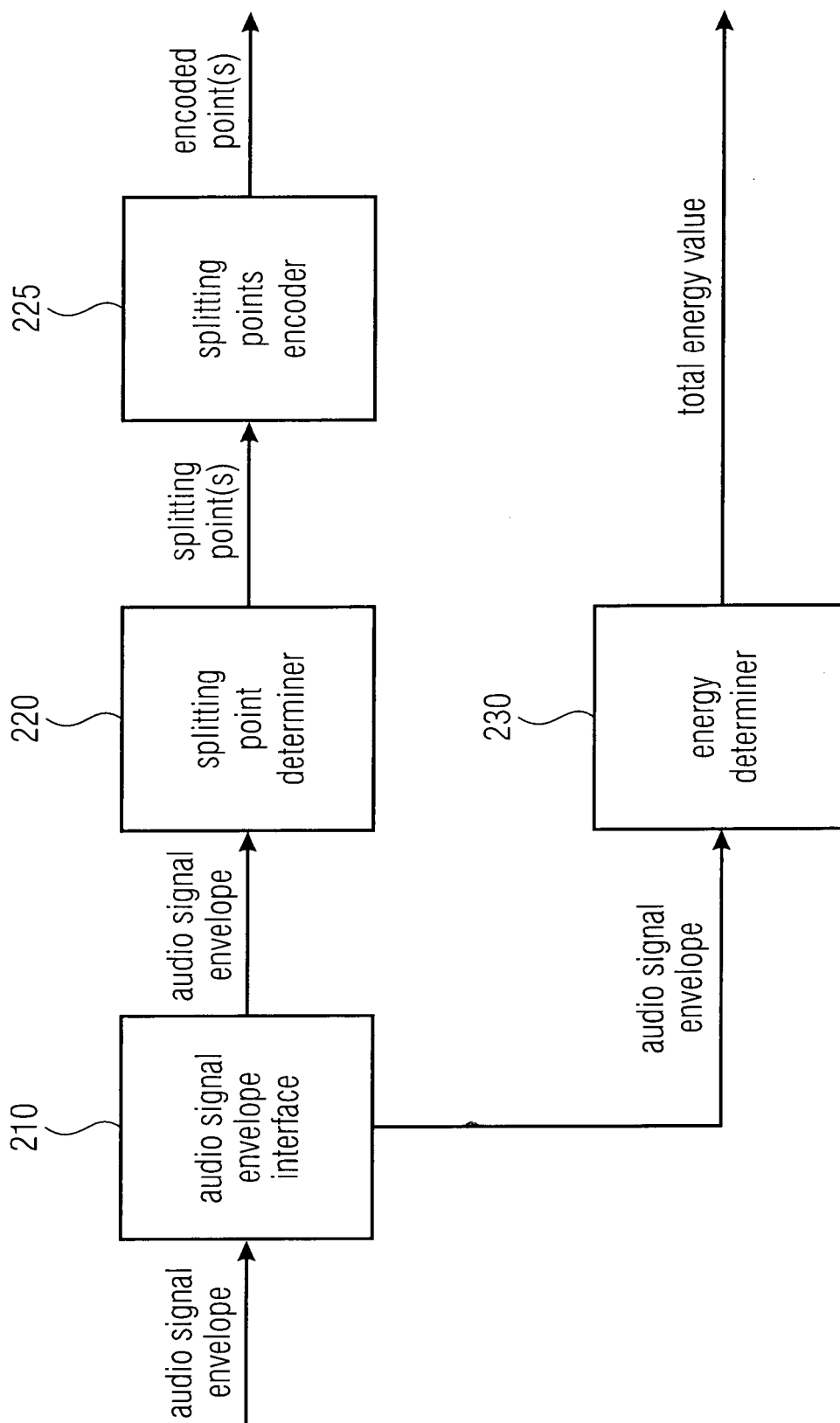


FIG 5

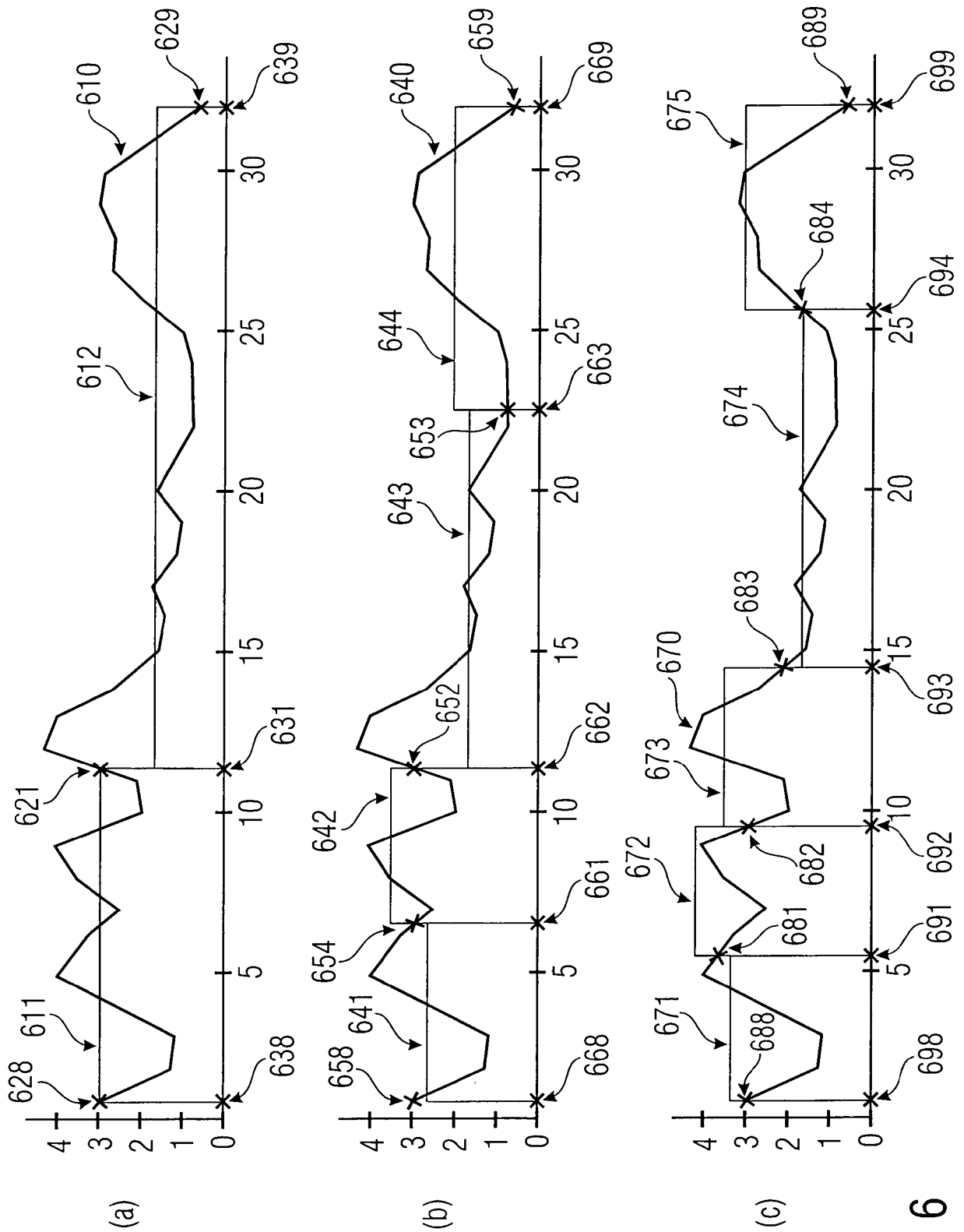


FIG 6

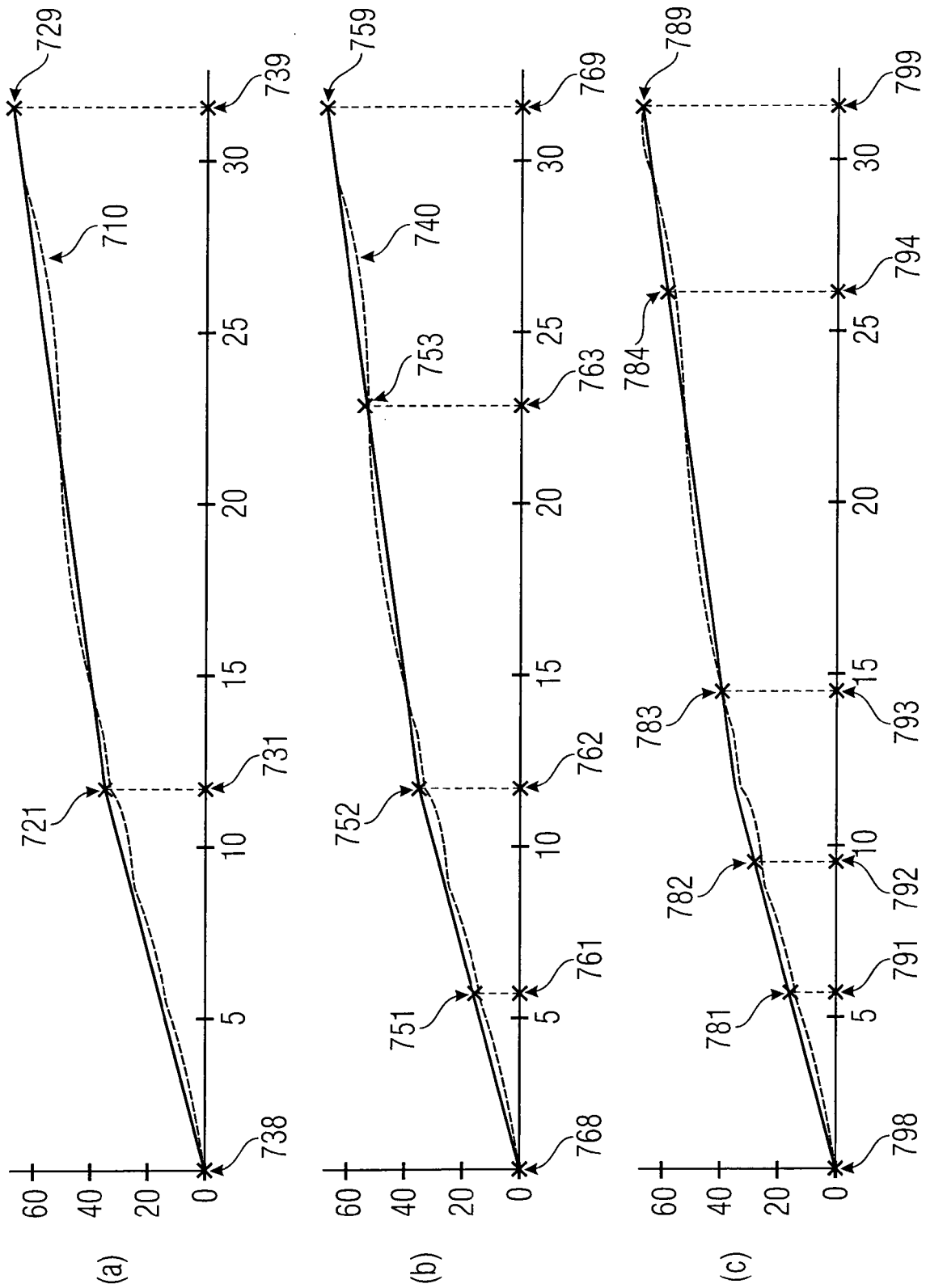


FIG 7

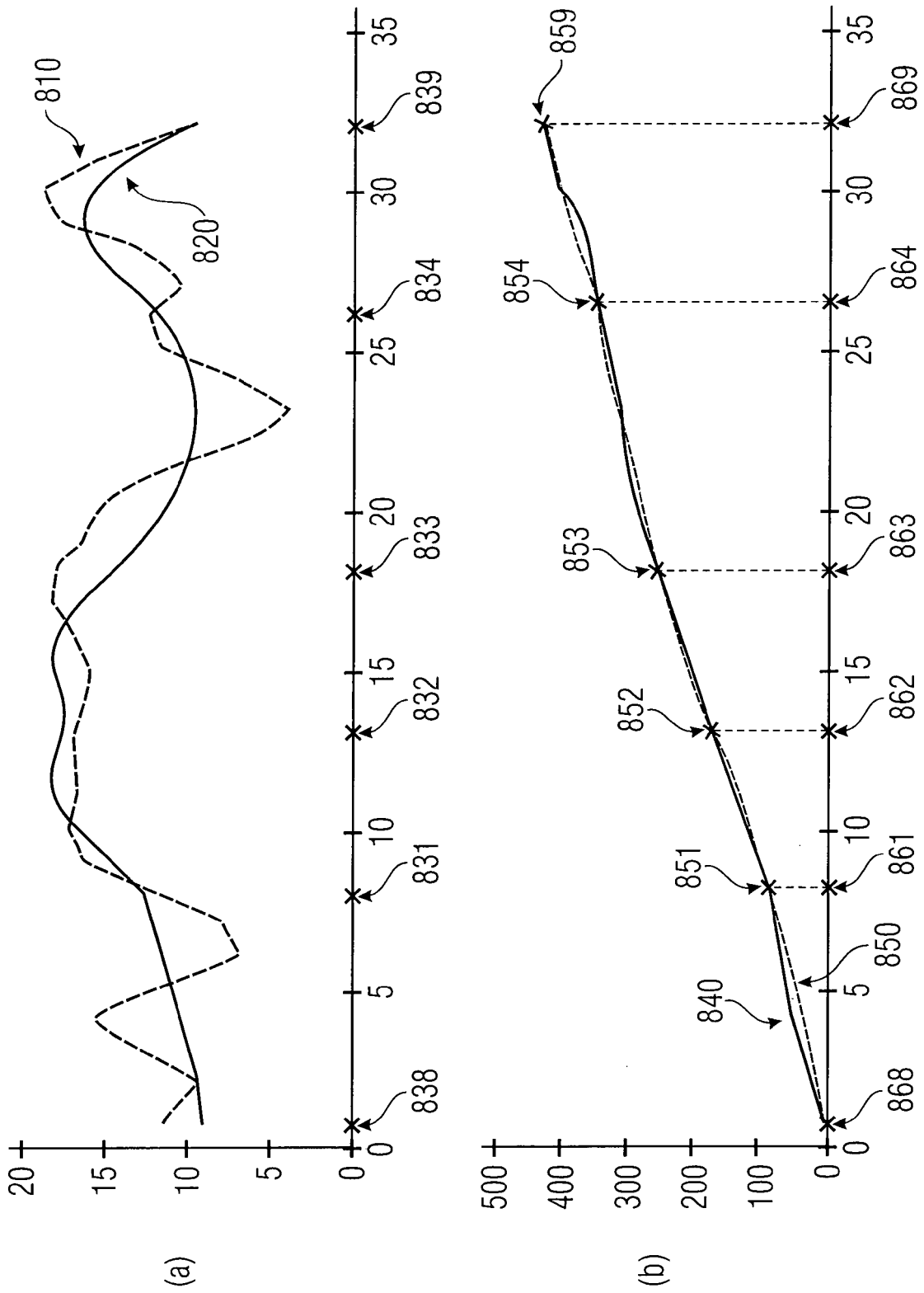


FIG 8

9/17

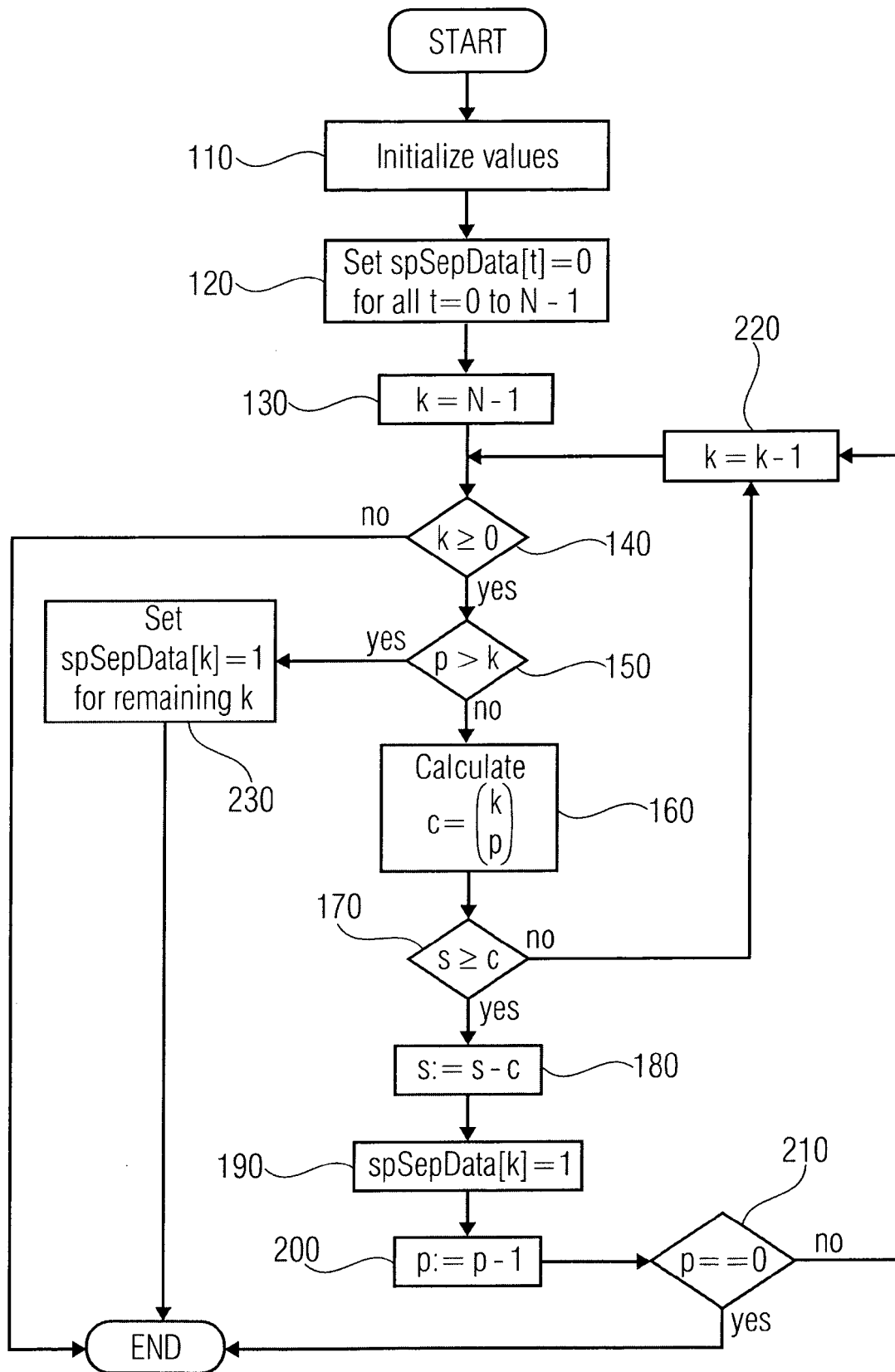


FIG 9

10/17

```
s = bsSpCodedPos;
p = bsSpNumTrSlots + 1;
N = numSlots

for (k=0; k<N; k++)
{
    spSepData[k]=0;
}

for (k=N-1; k>=0; k--)
{
    if (p > k) {
        for (;k>=0; k--)
            spSepData[k]=1;
        break;
    }
    c = k-p+1;
    for (h=2; h<=p; h++) {
        c *= k - p + h;
        c /= h;
    }
    if (s >= (int)c) { /* c is long long for up to 32 slots */
        s -= c;
        spSepData[k]=1;
        p--;
        if (p == 0)
            break;
    }
}
```

FIG 10

11/17

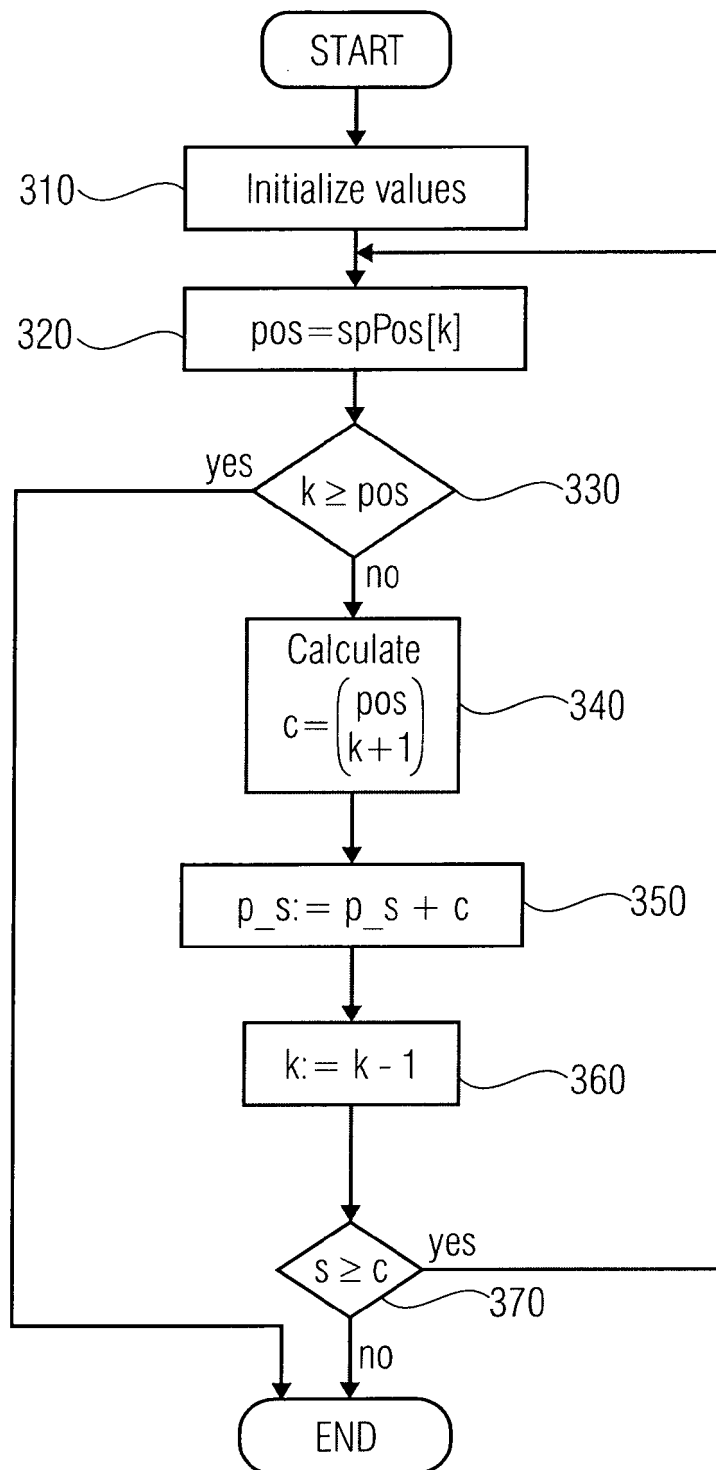


FIG 11

12/17

```

p_s = 0;
for (k=spPosLen-1; k>=0; k--){
    pos = spPos[k];
    if (k >= pos) {
        break;
    }
    c = 1; /* c is long long for up to 32 pos*/
    for (h=1; h<=(k+1); h++) {
        c *= pos-k-1 + h;
        c /= h;
    }
    p_s += c;
}
bsSpCodedPos = p_s;

```

FIG 12

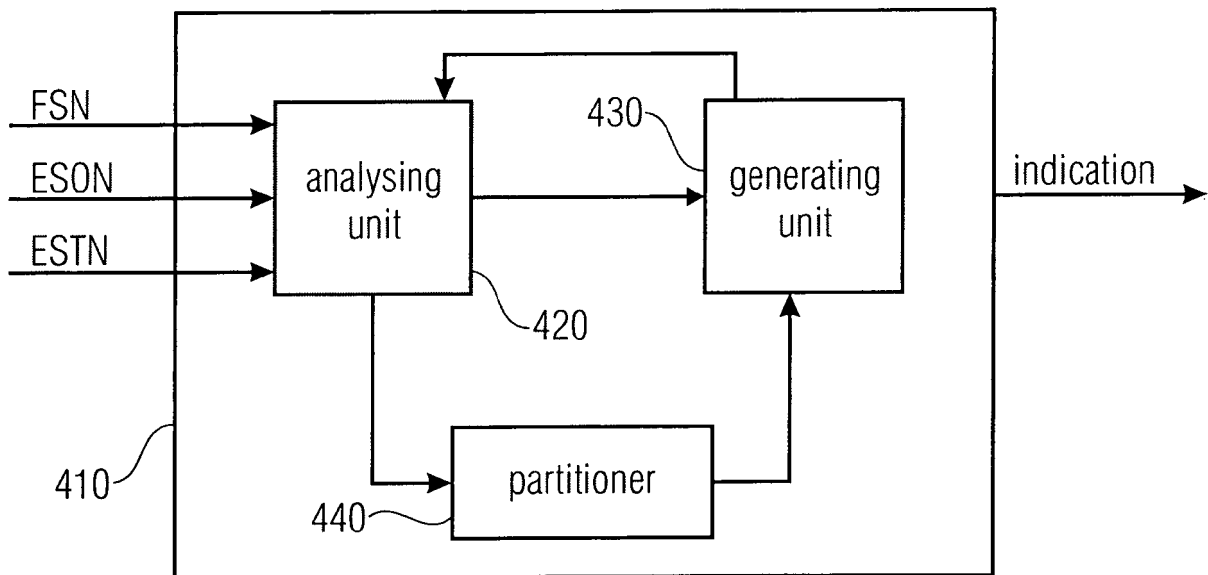


FIG 13

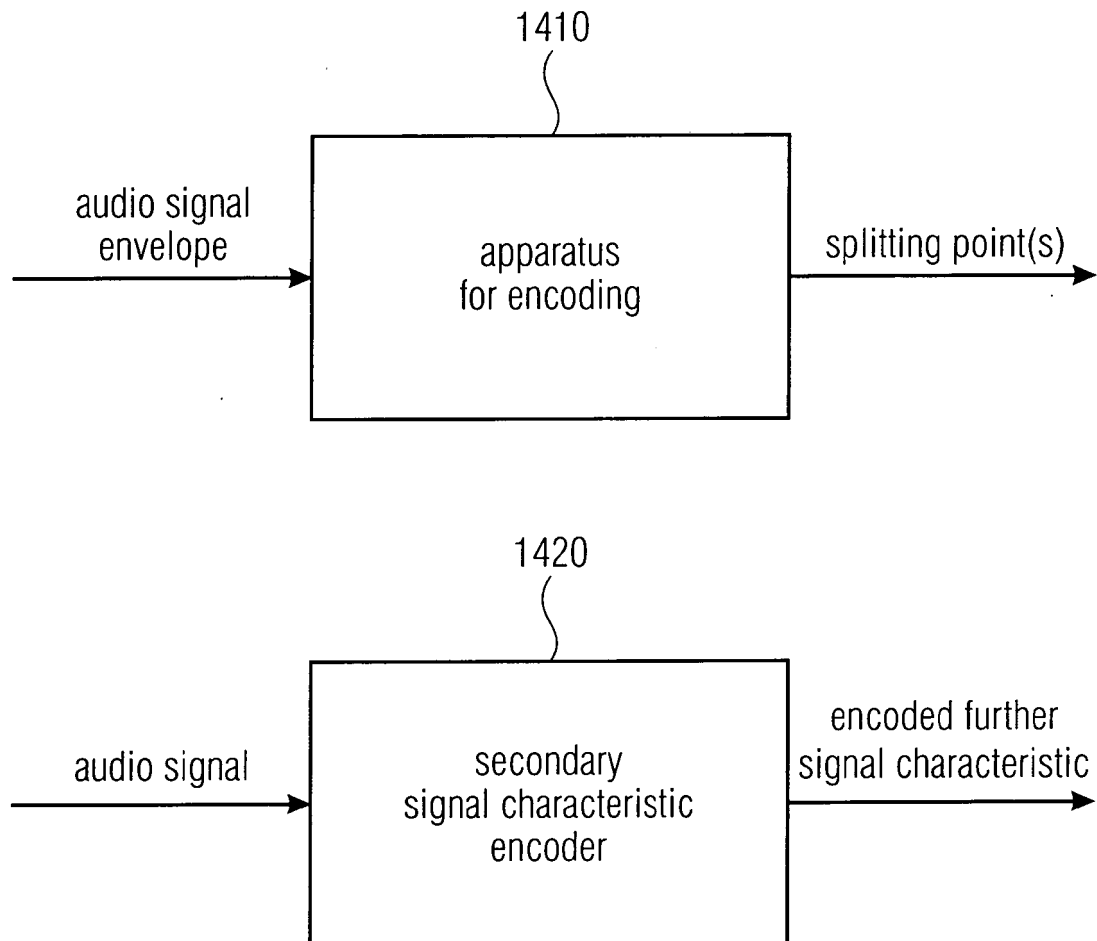


FIG 14

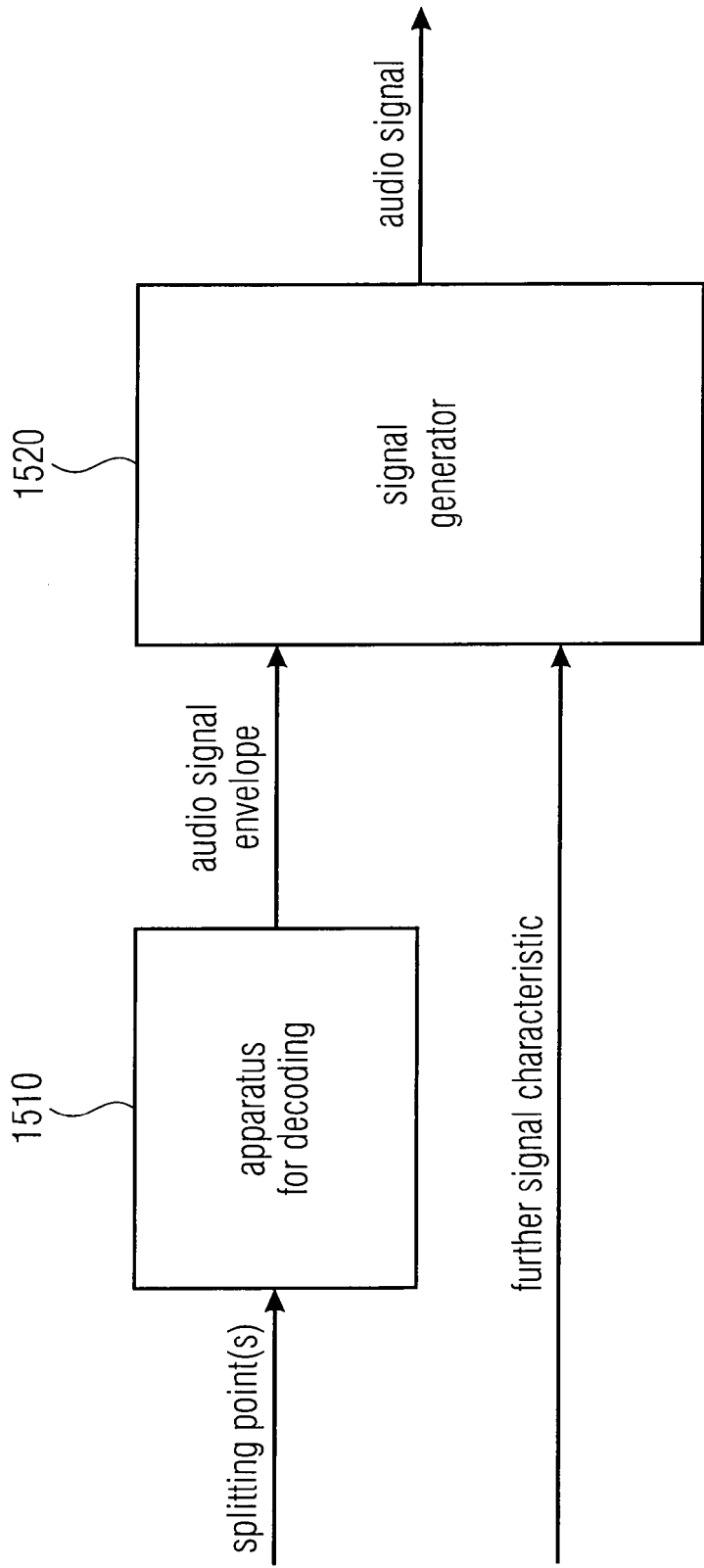


FIG 15

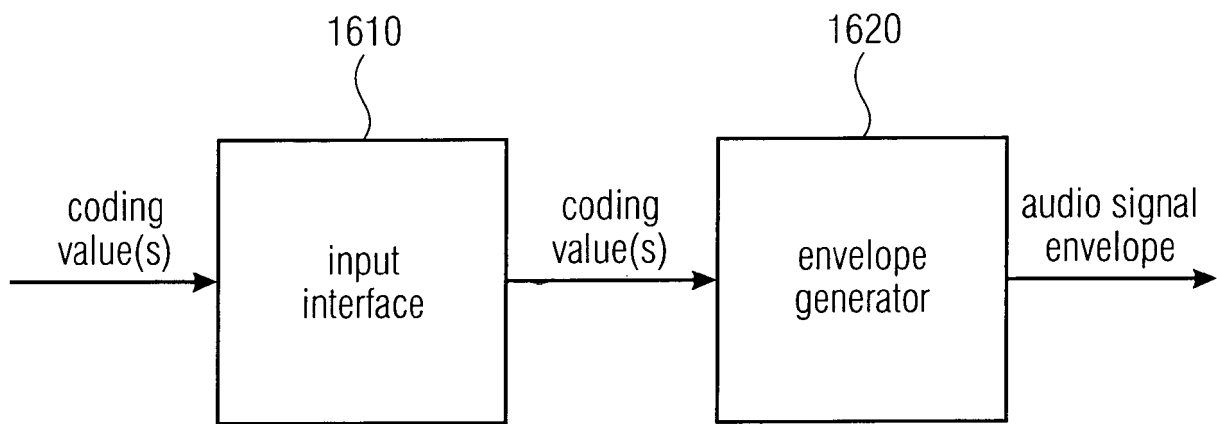


FIG 16

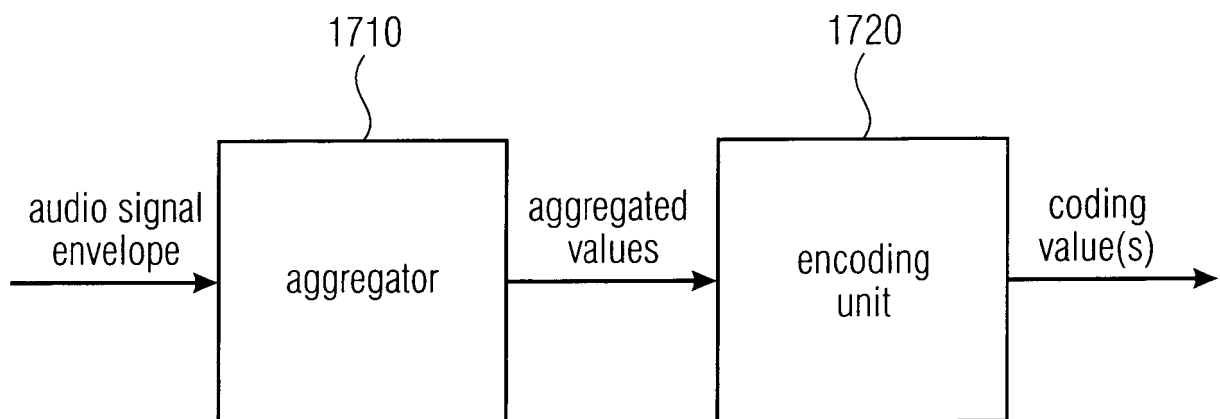


FIG 17

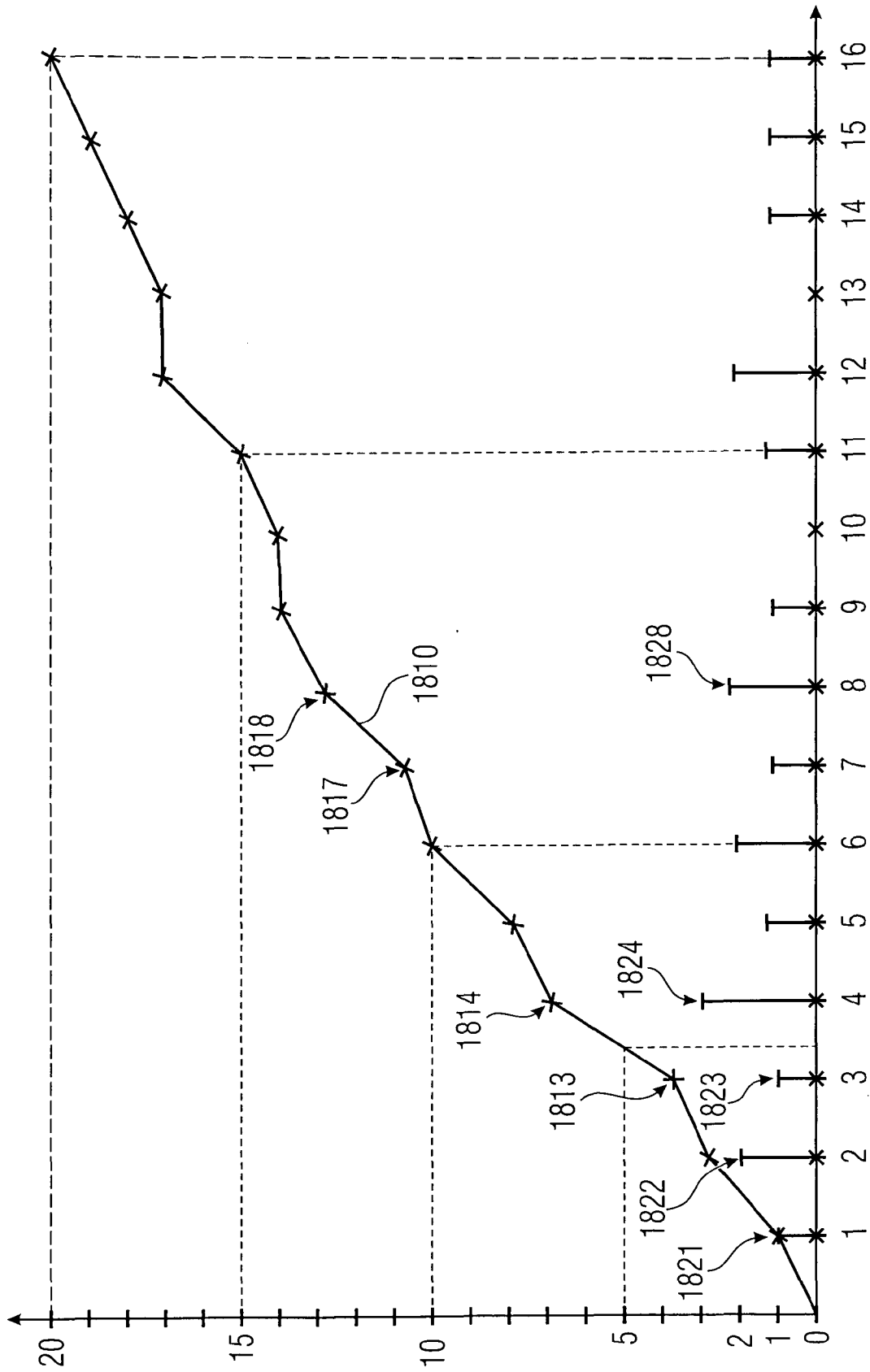


FIG 18

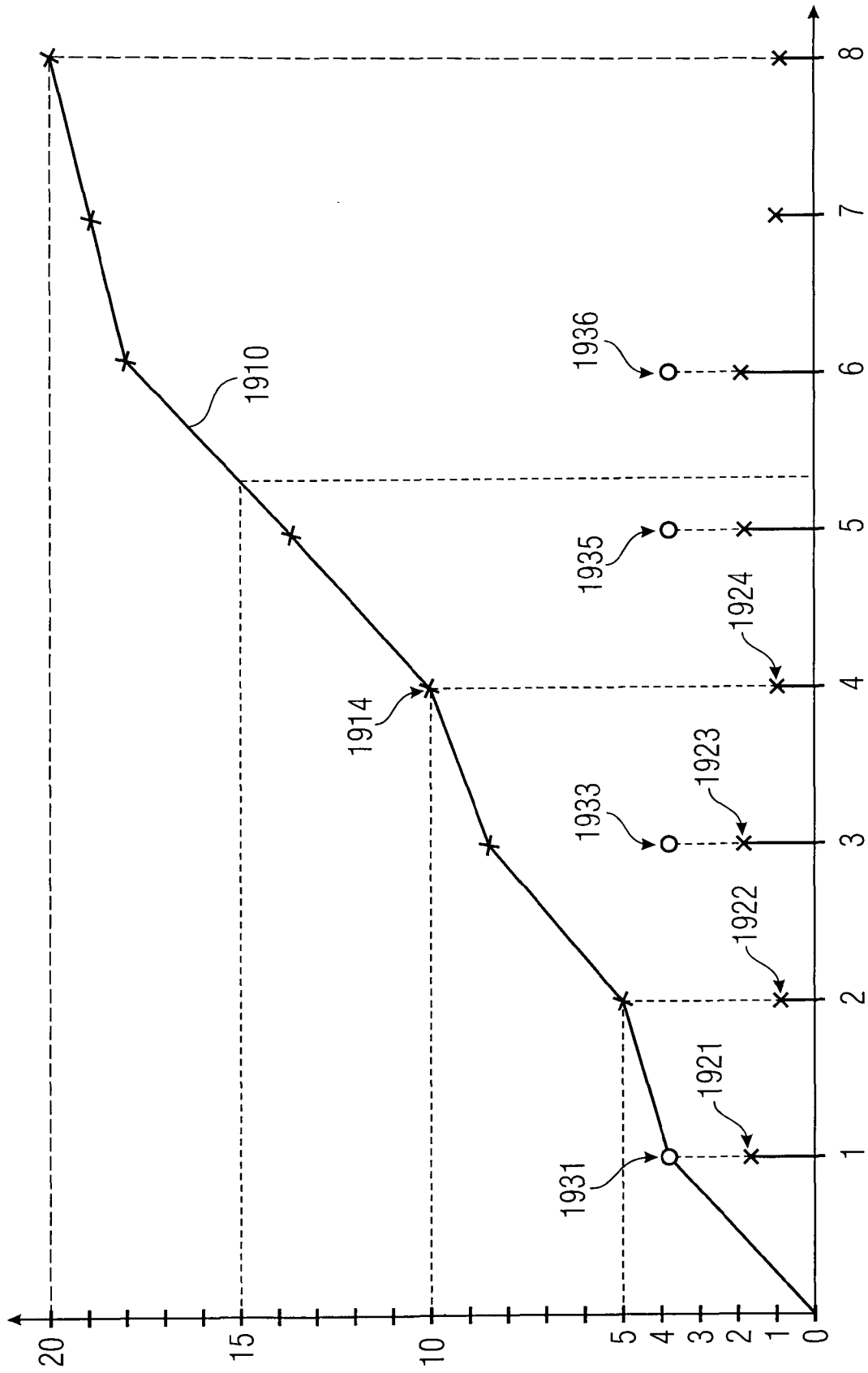


FIG 19