ارائه یک الگوریتم زمانبندی عادل برای سویچهای سلولی با صف ورودی[1]

رضا سعیدی نیا
کارشناسی ارشد دانشگاه علم و صنعت
r_saeidinia@mail.iust.ac.ir

محمد رضا سمیعی
کارشناسی ارشد دانشگاه آزاد
m_r_samiei@yahoo.com

چکیده :

در دنیای کنونی ارتباطات نقش عمده ای در زندگی انسان بر عهده دارد و از این میان می توان به اینترنت به عنوان یک شبکه جهانی و فراگیر اشاره کرد که رشد آن به صورت نمایی و روز افزون است . با افزایش تعداد کاربران در این شبکه بزرگ،ارائه راه حلهایی که بتواند مشکلات ترافیکی آنرا حل کند ضروری بنظر می رسد . برای این منظور چند راه حل وجود دارد که یکی از آنها استفاده از وسائل سوییچینگ سریع به منظور دستیابی به سرعت و کارایی بالا است . در طراحی مسیریاب های امروزی از ساختار سویچهای با صف ورودی استفاده می شود. هر چه تعداد پورتهای ورودی ، خروجی و سرعت این وسایل بیشتر می شود مساله زمانبندی در آنها بیشتر مورد توجه واقع می گیرد . در این مقاله یک الگوریتم زمانبندی برای ترافیک متحدالشکل ارائه شده است که دارای سرعت و عدالت بیشتر نسبت به سایر الگوریتمها می باشد . این الگوریتم برای زمانبندی از دو واحد طول سلول سر صف و تعداد سلولهای موجود در یک صف بهره می برد و سعی می کند عدالت را با واحد وزنی سن برقرار نماید و با استفاده از واحد تعداد سلول هر صف پورتهای با ترافیک بالا را نیز مد نظر قرار می دهد . براساس نتایج حاصل از شبیه سازی درمحیط Sim[1] الگوریتم بیشنهادی در مقایسه با سایر الگوریتم های موجود دارای کارایی بهتر در تاخیر و تغییرات تاخیر می باشد.

کلمات کلیدی : زمانبندی - کراسبار - صف خروجی مجازی - کیفیت سرویس - مسیریابی

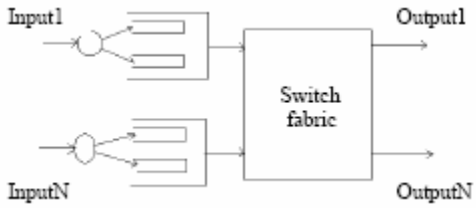
۱- مقدمه

امروزه ساختار سوییچهای سلولی با صف ورودی با ساختار کراسبار [۲] و تکنیک صف خروجی مجازی [۳] در مسیریابهای با سرعت بالامورد استفاده قرار می کیرد . علت استفاده از این پیکربندی ، انعطاف پذیری آن و امکان برقراری n ارتباط بین پورتهای ورودی و خروجی در آن واحد و همچنین عدم احتیاج به افزایش سرعت سوییچ فابریک نسبت به سرعت خط است . در سوییچهای با صفهای خروجی ، سوییچ فابریک باید N برابر سریعتر از سرعت خط عمل کند و همچنین سرعت حافظه های موجود در پورتهای خروجی نیز باید N+1 برابر سرعت خط باشد و با توجه به اینکه زمان دسترسی به حافظه ها در مقایسه با افزایش سرعت واسطهای شبکه ، رشد

[1] -Input Queued Cell Switches
[2] -Crossbar
[3] -Virtual Output Queueing

زیادی نداشته است ، از سوییچهای با صف خروجی زیاد در ساختار مسیریابهای شبکه استفاده نمی شود . سوییچهای با صف ورودی نیز از مساله بلاک شدن بسته های یک صف ورودی پشت سلول سر صف [1] رنج میبرند که با ارایه راه حل صف خروجی مجازی این مساله حل میشود [2,3,4] .

الگوریتمهای زمانبندی در هر برش زمانی [2] از بین مجموعه ورودی ها و خروجی ها ، تطابقی [3] را پیدا می کنند به نحوی که در این تطابق حداکثر یک خروجی به یک ورودی وصل شده باشد . پیدا کردن تطابقی که حداکثر وروردیها را به خروجی ها وصل کند دارای پیچیدگی زمانی بالایی است [5,6,7]. شکل کلی یک سویچ با صف ورودی درشکل ۱ نشان داده شده است . همانطور که دراین شکل دیده می شود ، Nصف در ورودی سوییچ (سمت چپ) و N صف در خروجی سوییچ قرار دارند . در هر پورت ورودی ، N صف وجود دارد که به هر کدام صف خروجی مجازی[4] گفته می شود و با VOQ_{ij} نشان داده می شوند و نشان دهنده صف موجود در پورت ورودی iام برای پورت خروجی jام هستند .



شکل ۱- ساختار سوییچ با صف ورودی و صف خروجی مجازی

در این ساختار مسئله HOL به طور کامل حل می شود زیرا هر سلول که بخواهد به یک خروجی فرستاده شود در VOQ مربوط به آن خروجی قرار می گیرد . زمانبند تطابق را بین صفهای VOQ وسوییچ فابریک برقرار می کند . از ساختار فوق در طراحی مسیریابهای با سرعت بالا استفاده می شود .

۲- الگوریتم پیشنهادی

برای توضیح این الگوریتم تعاریف زیر بیان شده است[7] :

۲-۱- ماتریس وزن : این ماتریس یک ماتریس N*N است که عناصر آن نشان دهنده وزنهای وابسته به ورودیها و خروجی های مربوطه است سطرهای این ماتریس مشخص کننده ورودی ها و ستونهای آن مشخص کننده خروجی ها هستند. در الگوریتم ارائه شده وزن (Wij) متشکل از مجموع دو کمیت است که عبارتند از :

- تعداد سلولهای موجود در یک صف ورودی برای یک پورت خروجی خاص (Lij)
- سن یا مدت زمان انتظار سلول سر صف در یک صف ورودی برای پورت خروجی خاص (Aij) یعنی :

$$Wij = Lij + Aij \qquad (۱)$$

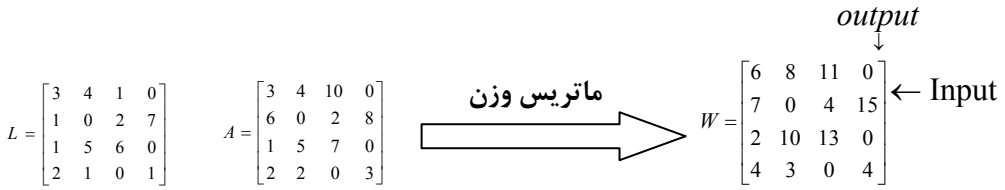
برای روشن شدن مطلب مثال زیر آورده شده است :

اگر ماتریس تعداد سلولها (L) و ماتریس سن یا مدت زمان انتظار (A) به صورت زیر باشند :

---

[1] -Head Of Line Blocking
[2] -Time Slice
[3] -Match
[4] Virtual Output Queuing(VOQ)

به صورت زیر باشد :



$$L=\begin{bmatrix}3&4&1&0\\1&0&2&7\\1&5&6&0\\2&1&0&1\end{bmatrix}\quad A=\begin{bmatrix}3&4&10&0\\6&0&2&8\\1&5&7&0\\2&2&0&3\end{bmatrix}\quad W=\begin{bmatrix}6&8&11&0\\7&0&4&15\\2&10&13&0\\4&3&0&4\end{bmatrix}$$

همانطور که مشاهده می شود در ماتریس وزن درایه (4,2) برابر ۳ می باشد چرا که در پورت ورودی ۴ برای پورت خروجی ۲ ، یک سلول وجود داشته است که این سلول ۲ قطعه زمانی نیز منتظر شده است ولی هنوز توسط زمانبند به آن سرویس داده نشده است.

۲-۲- ماتریس تطابق : ماتریس N*N است که عناصر آن فقط ۱ یا صفر می باشد و در هر سطر و ستون آن فقط یک عنصر غیر صفر وجود دارد . وجود یک عنصر ۱ در یک سطر و ستون به معنی بر قراری اتصال بین آن دو می باشد. مثلا :

$$M=\begin{bmatrix}0&0&1&0\\1&0&0&0\\0&1&0&0\\1&0&0&1\end{bmatrix}$$

در این ماتریس همانطور که مشاهده می شود درایه واقع در سطر ۲ ستون ۱ برابر یک می باشد یعنی بین پورت ورودی ۲ و خروجی ۱ ، یک تطابق برقرار شده است. این ماتریس یک ماتریس MSM است زیرا در یک قطعه زمانی حداکثر تطابق ممکن را پیدا کرده است .

الگوریتم ارائه شده که **WFMC** نامگذاری شده است به این صورت عمل می کند که روی ستونهای ماتریس وزن تمرکز می کند و بزرگترین عنصر واقع در ستونهای ماتریس را پیدا می کند ، بطوریکه برخوردی بین آنها وجود نداشته باشد و بر این اساس ماتریس تطابق را می سازد. همانطور که قبلا نیز گفته شد درایه های ماتریس وزن از مجموع دو کمیت یعنی تعداد سلول منتظر در یک صف ورودی و سن (مدت زمان انتظار) سلولهای سر صف ، تشکیل شده است . در نظر گرفتن این کمیت باعث می شود که این الگوریتم نسبت به الگوریتمهایی مثل LQF ، ILQF ، ISLIP ، OCF ، IOCF [7,8]و... عدالت بیشتری داشته باشد وتا جای ممکن از بروز قحطی زدگی [1] دریک صف جلوگیری می کند.

دلیل آن اینست که در الگوریتمهایی مثل LQF ، ILQF ، ISLIP که واحد وزنی در آنها طول صف (تعداد سلول منتظر در یک صف) می باشد ، صفهایی که ترافیک ورودی آنها کم است بعد از گذشت مدت زمانی به احتمال زیاد در حالت انتظار زیادی می مانند و دچار قحطی زدگی می شوند[7] . همچنین الگوریتمهایی نظیر OCF و IOCF که واحد وزنی آنها سن (مدت زمان انتظار) سلولهای سر صف می باشد ، اولویت را به سلولهایی می دهند که زیاد در صف منتظر مانده اند و بنابراین به عنوان مثال اگر یک صف دارای ۱۰سلول باشد و در هر قطعه زمانی نیز یک سلول به آن وارد شود و در نتیجه دارای سن ۱۰ نیز باشد و صف دیگری ۱ سلول داشته باشد که در ۱۰ قطعه زمانی سرویسی نگرفته باشد و در نتیجه دارای سن ۱۱ باشد (ترافیک آن کم است) ، الگوریتم OCF یا IOCF سلول با سن ۱۱ را انتخاب کرده و آنرا به خروجی می‌فرستد که این دارای اشکال است، چرا که به هیچ وجه ترافیک بالای یک صف را در نظر نمی‌گیرند. در حالیکه در الگوریتم ارائه شده WFMC علاوه بر سن سلولهای واقع در یک صف، ترافیک ورودی به یک صف نیز در نظر گرفته می‌شود. یعنی در مثال فوق ماتریس وزن که به ترتیب W_1 =۱۰+۱۰=۲۰ و W_2=۱۱+۱=۱۲ می‌باشند، مطابق با الگوریتم **WFMC** اولویت به صف با ترافیک بالاتر داده می‌شود (W_1)که البته صف دوم نیز حتما بعد از ۲۰ قطعه زمانی سرویس می‌گیرد و از بروز قحطی زدگی تا حد زیادی جلوگیری می‌شود .

[1] -starvation

الگوریتم **WFMC** دارای سه گام است که عبارتند از:

گام اول : محاسبه ماتریس وزنی : پیدا کردن وزنهای یالهای گراف دوقطبی در این گام انجام می شود . این گامی است که دارای سربار نسبتا زیادی است . $W_{ij} = L_{ij} + A_{ij}$

که در آن L_{ij} تعداد سلول پورت ورودی iام برای پورت خروجی j می باشد. و A_{ij} سن سلول سر صف می باشد.

گام دوم : محاسبه ماتریس تطابق : مهمترین گام الگوریتم می باشد که باید از ماتریس وزن ماتریسی پیدا کند که فقط از درایه های۱و۰ تشکیل شده است و حداکثر دارای n (تعداد پورتهای سویچ) درایه ۱ است زیرا در آن واحد نمی توان از یک پورت ورودی دو بسته به یک پورت خروجی ارسال کرد . همچنین یک پورت خروجی در آن واحد نمی تواند بیشتر ازیک بسته دریافت کند . WFMC بزرگترین عنصر در هر ستون را انتخاب می‌کند (ستونهایی را پیدا می کندکه دارای بزرگترین واحد وزنی باشند)

گام سوم : پیکربندی سویچ بر اساس ماتریس تطابق پیدا شده : چون ساختار سویچ کراسبار است و می تواند در آن واحد n سلول را بفرستد بنابراین این گام بصورت سخت افزاری پیاده سازی می شود و سریع است .

شبه کد الگوریتم درجدول زیر آمده است :

جدول ۱- شبه کد مربوط به الگوریتم WFMC

```
WFMC Algorithm /find maximum of columns in weight matrix
                Whitch is combination of length matrix and age matrix
1. flag [i] = 0    ,        ∀ i = 1 , … , N
   weight [i] [j] = length [i] [j] + age [i] [j] , ∀ i, j =1, …,N
2. match [i] [j] = 0        ,        ∀ i, j = 1 , …. , N
3. for (j = ϕ ; j < n ; j++)
maximum = weight [0 ] [j] ;
t1 = 0
for (i = ϕ ; i < n1 ; i++)
{
if (flag [i] = = 0  &&  maximum < weight [i] [j])
   {
t1 = i ;
maximum = weight [i] [j]
   }
}
if (weight [t1] [j] = 1
   {
match [t1] [j] = 1
flag [t1 ] =1
   }
4. configure the fabric based on the match matrix
```

پیچیدگی محاسباتی این الگوریتم در حالت متوسط از $O(n^2)$ می باشد . به عنوان نمونه الگوریتم با مثال زیر توضیح داده شده است :

فرض می کنیم ماتریس وزن (حاصل جمع دو ماتریس تعداد سلولها و سن سلول ها) به صورت زیر باشد .

$$W = \begin{bmatrix} 6 & 8 & 11 & 0 \\ 7 & 0 & 4 & 15 \\ 2 & 10 & 13 & 0 \\ 4 & 3 & 0 & 11 \end{bmatrix}$$

نتیجه اجرای الگوریتم به صورت ⟹

$$match = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

همانطور که مشاهده می شود در ستون اول : عنصر بزرگتر یعنی ۷ واقع در سطر ۲ انتخاب می شود و **match [2] [1]=1** می شود . در ستون دوم : عنصر بزرگتر یعنی ۱۰ واقع در سطر ۳ انتخاب می شود و **match [3] [2] =1** می شود. در ستون سوم : عنصر بزرگتر یعنی ۱۳ باید انتخاب شود که چون قبلا در همان سطر ۱۰ انتخاب شده است بنابراین بجای ۱۳، عنصر واقع در سطر ۱ یعنی ۱۱

انتخاب می شود و **match [1][3]=1** می شود. در ستون چهارم : عنصر واقع در سطر دوم یعنی ۱۵ باید انتخاب شود که مجددا از آنجایی که قبلا عنصر ۷ واقع در این سطر انتخاب شده است بجای ۱۵ عدد بزرگتر بعدی یعنی ۱۱ انتخاب می شود و **match [4][4]=1** می شود . وزن تطابق پیدا شده برابر با : **w= 7+10+11+11=39** است .

الگوریتم **WFMC** درمقایسه با سایر الگوریتمها زمان بهتری را دارد و تا حد زیادی به سمت MSM میل می کند و با MWM نیز فاصله چندانی ندارد . (وزن الگوریتم MWM با توجه به این مثال **W= 7+10+13+15=45** می باشد)

۳- شبیه سازی

برای شبیه سازی الگوریتم WFMC از دو پارامتر میانگین تاخیرات روی کل سلولها و انحراف معیار تاخیرات که نشان دهنده جیتر می باشد و برای کاربردهای بلادرنگ[1] مفید است ، استفاده شده است و فرض شده که ترافیک ورودی به سویچ از نوع متحدالشکل است . نتایج برای سویچ ۳۲×۳۲ پیاده سازی شده و شبیه سازی با فرض ورود ۲۰۰۰۰۰، ۵۰۰۰۰۰ و ۱۰۰۰۰۰۰ سلول به سوییچ انجام شده است . این سلولها با توزیع برنولی به سوییچ وارد میشوند . همچنین برای الگوریتمهای تکراری نظیر ISLIP، ILQF و IOCF شبیه سازی به ازای تعداد تکرار (i=2 و i=5) انجام شده است که به ازای i=5 بهترین جواب برای یک الگوریتم تکراری بدست می آید .شبیه سازی با نرم افزار SIM طراحی شده در دانشگاه استانفورد انجام شده است که یک نرم افزار بسیار قوی خاص شبیه سازی سوییچ های ATM با سرعت بالا و نوشته شده به زبان ANSI C است که تحت لینوکس قابل اجرا می باشد .

جدول ۲-۱- مقایسه میانگین تاخیرات الگوریتمهای مختلف به ازای ورود ۲۰۰۰۰۰ بسته به سویچ و تعداد تکرار (I=2)

Delay	ISLIP	ILQF	IOCF	WFMC	ILPF	IOPF
0.1	0.0593	0.0593	0.0598	0.0594	0.0606	0.0603
0.2	0.1453	0.1445	0.148	0.1455	0.1562	0.1535
0.3	0.2708	0.2689	0.2815	0.2708	0.318	0.3043
0.4	0.4559	0.4517	0.4864	0.4534	0.593	0.5487
0.5	0.7421	0.7287	0.8078	0.7299	1.0204	0.9225
0.6	1.22	1.1901	1.3468	1.1645	1.6826	1.5046
0.7	2.1574	2.0542	2.3355	1.9108	2.7909	2.4952
0.8	5.2349	4.4637	4.7686	3.4128	5.0926	4.6391
0.9	54.3976	2402.798	4326.25	7.9541	127713.01	12.8579
0.97	174.5783	12691.54	14225.83	27.0991	209554.254	106.6599

جدول ۲-۲- مقایسه انحراف معیار تاخیرات الگوریتمهای مختلف به ازای ورود ۲۰۰۰۰۰ بسته به سویچ و تعداد تکرار (I=2)

Standard Deviation	ISLIP	ILQF	IOCF	WFMC	ILPF	IOPF
0.1	0.2636	0.2683	0.2459	0.2551	0.2769	0.2481
0.2	0.4423	0.4581	0.3857	0.4129	0.4874	0.3992
0.3	0.6444	0.6821	0.5322	0.5815	0.7536	0.5723
0.4	0.8948	0.9694	0.6964	0.7707	1.0929	0.7873
0.5	1.2466	1.3702	0.9144	1.0185	1.5643	1.1005
0.6	1.7963	2	1.2357	1.3667	2.2916	1.6488
0.7	2.8689	3.1643	1.768	1.9165	3.5034	2.6705
0.8	6.7489	6.3744	2.8473	2.9661	6.0046	4.9791
0.9	57.0814	536.6956	832.654	6.3343	24758.98	14.3915
0.97	178.6558	2521.233	2738.127	26.2369	68249.1453	127.3391

[1] Real Time

جدول ۲-۲- مقایسه میانگین تاخیرات الگوریتمهای مختلف
به ازای ورود ۲۰۰۰۰۰ بسته به سوییچ و تعداد تکرار (I=5)

Delay	ISLIP	ILQF	IOCF	WFMC	ILPF	IOPF
0.1	0.0593	0.0593	0.0598	0.0594	0.0606	0.0603
0.2	0.1453	0.1445	0.148	0.1455	0.1562	0.1535
0.3	0.2708	0.2689	0.2815	0.2708	0.318	0.3043
0.4	0.4557	0.4506	0.4864	0.4534	0.593	0.5487
0.5	0.7408	0.7287	0.8069	0.7299	1.0204	0.9225
0.6	1.2082	1.1773	1.3463	1.1645	1.6826	1.5046
0.7	2.0431	1.963	2.3108	1.9108	2.7909	2.4952
0.8	3.809	3.5783	4.3742	3.4128	5.0926	4.6391
0.9	9.7249	8.4912	11.0414	7.9541	127713.01	12.8579
0.97	42.7784	28.0596	41.0636	27.0991	250236.32	106.6599

جدول ۲-۴- مقایسه انحراف معیار تاخیرات الگوریتمهای مختلف
به ازای ورود ۲۰۰۰۰۰ بسته به سوییچ و تعداد تکرار (I=5)

Standard Deviation	ISLIP	ILQF	IOCF	WFMC	ILPF	IOPF
0.1	0.2636	0.2683	0.2459	0.2551	0.2769	0.2481
0.2	0.4423	0.4581	0.3857	0.4129	0.4874	0.3992
0.3	0.6444	0.6821	0.5322	0.5815	0.7536	0.5723
0.4	0.8943	0.9676	0.6963	0.7707	1.0929	0.7873
0.5	1.2435	1.3667	0.9138	1.0185	1.5643	1.1005
0.6	1.7757	1.9741	1.2356	1.3667	2.2916	1.6488
0.7	2.6878	2.9955	1.7544	1.9165	3.5034	2.6705
0.8	4.5849	5.0177	2.7373	2.9661	6.0046	4.9791
0.9	11.1787	10.9808	5.7213	6.3343	24758.98	14.3915
0.97	47.8822	29.5637	17.2151	26.2369	68452.26	127.3391

جدول۳-۱- مقایسه میانگین تاخیرات الگوریتمهای مختلف
به ازای ورود ۱۰۰۰۰۰۰ بسته به سوییچ و تعداد تکرار (I=2)

Delay	ISLIP	ILQF	IOCF	WFMC	ILPF	IOPF
0.1	0.0588	0.0588	0.0591	0.0588	0.0598	0.0595
0.2	0.145	0.1444	0.1477	0.1451	0.1558	0.1527
0.3	0.27	0.2685	0.2812	0.2699	0.3174	0.3035
0.4	0.4564	0.4512	0.487	0.4545	0.5922	0.5481
0.5	0.7442	0.7309	0.8104	0.7313	1.0227	0.9249
0.6	1.2234	1.1905	1.3491	1.1673	1.6837	1.5041
0.7	2.1551	2.062	2.3453	1.9113	2.7994	2.5025
0.8	5.2594	4.4886	4.7922	3.4263	5.1148	4.6614
0.9	54.6755	11845.14	21570.56	7.951	25412.156	12.8318
0.97	178.1801	33524.12	65234.21	27.7662	152321.16	110.4259

جدول ۳-۲- مقایسه انحراف معیار تاخیرات الگوریتمهای مختلف
به ازای ورود ۱۰۰۰۰۰۰ بسته به سوییچ و تعداد تکرار (I=2)

Standard Deviation	ISLIP	ILQF	IOCF	WFMC	ILPF	IOPF
0.1	0.2632	0.2675	0.2446	0.2545	0.2747	0.2463
0.2	0.4417	0.4578	0.3858	0.4131	0.4873	0.3988
0.3	0.6426	0.68	0.5302	0.5782	0.7507	0.57
0.4	0.8968	0.9685	0.6988	0.7735	1.0941	0.7884
0.5	1.2485	1.3729	0.918	1.0211	1.565	1.1053
0.6	1.8038	2.0097	1.2404	1.3698	2.2974	1.653
0.7	2.8697	3.1788	1.7712	1.9172	3.5171	2.6795
0.8	6.7672	6.4016	2.8617	2.9739	6.0335	5.0072
0.9	57.449	2369.339	4142.743	6.2654	6528.26	14.3262
0.97	183.6086	10236.15	12354.14	25.6953	20358.21	1131.917

جدول ۳-۲- مقایسه انحراف معیار تاخیرات الگوریتمهای مختلف
به ازای ورود ۱۰۰۰۰۰۰ بسته به سوییچ و تعداد تکرار (I=5)

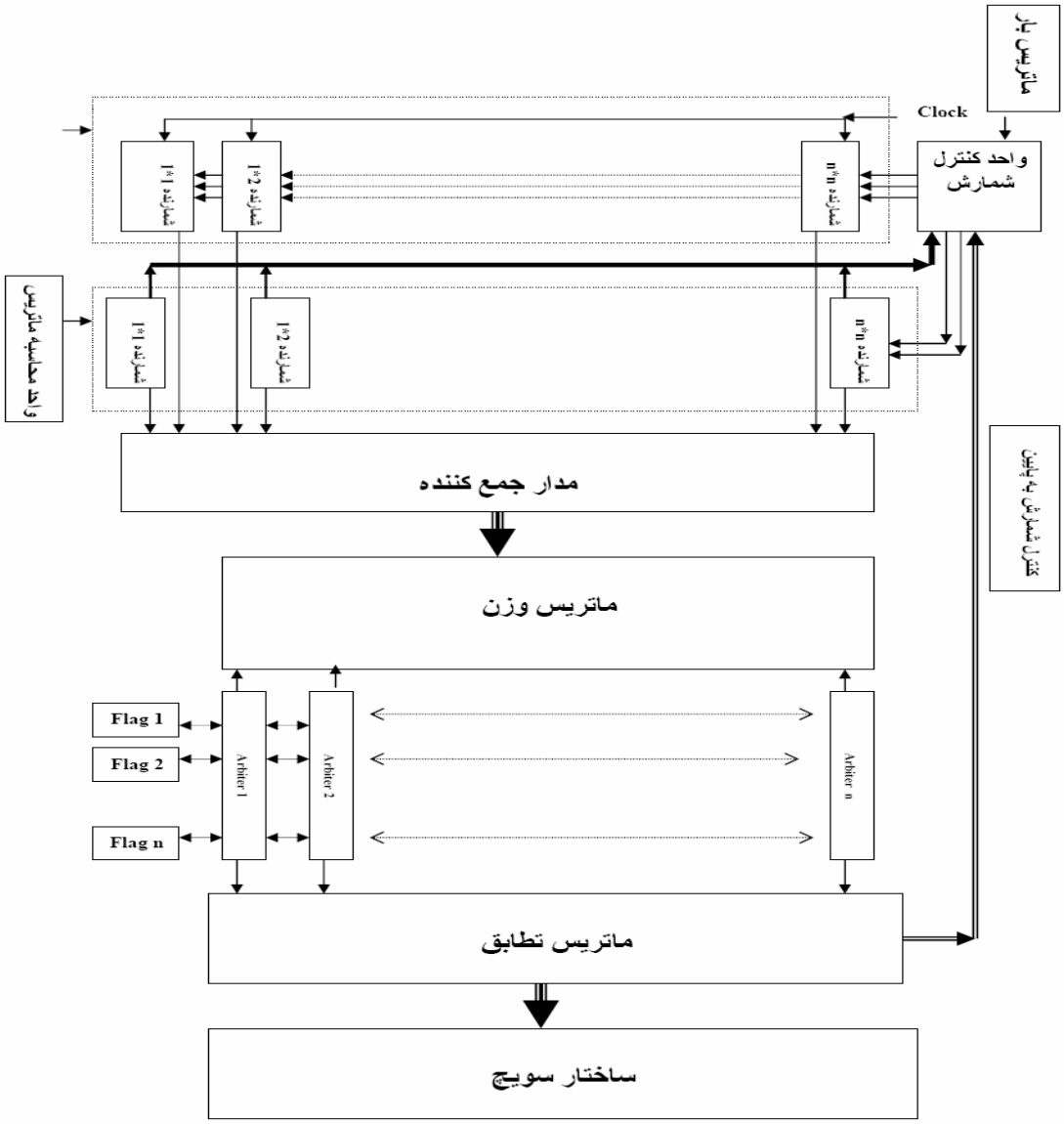
Standard Deviation	ISLIP	ILQF	IOCF	WFMC	ILPF	IOPF
0.1	0.2632	0.2675	0.2446	0.2545	0.2747	0.2463
0.2	0.4417	0.4578	0.3858	0.4131	0.4873	0.3988
0.3	0.6426	0.6800	0.5302	0.5782	0.7507	0.5700
0.4	0.8964	0.9679	0.6989	0.7735	1.0941	0.7884
0.5	1.2453	0.3695	0.9177	1.0211	1.5650	1.1053
0.6	1.7788	1.9824	1.2383	1.3698	2.2974	1.6530
0.7	2.6904	3.0026	1.7568	1.9172	3.5171	2.6795
0.8	4.6044	5.0292	2.7538	2.9739	6.0335	5.0072
0.9	11.1475	10.9453	5.6446	6.2654	High	14.3262
0.97	48.3046	30.1896	17.8289	25.6953	High	131.9171

جدول۳-۳- مقایسه میانگین تاخیرات الگوریتمهای مختلف
به ازای ورود ۱۰۰۰۰۰۰ بسته به سوییچ و تعداد تکرار (I=5)

Delay	ISLIP	ILQF	IOCF	WFMC	ILPF	IOPF
0.1	0.0588	0.0588	0.0591	0.0588	0.0598	0.0595
0.2	0.1450	0.1444	0.1477	0.1451	0.1558	0.1527
0.3	0.2700	0.2685	0.2812	0.2699	0.3174	0.3035
0.4	0.4563	0.4508	0.4870	0.4545	0.5922	0.5481
0.5	0.7426	0.7294	0.8103	0.7313	1.0227	0.9249
0.6	1.2098	1.1783	1.3456	1.1673	1.6837	1.5041
0.7	2.0433	1.9653	2.3152	1.9113	2.7994	2.5025
0.8	3.8250	3.5879	4.3899	3.4263	5.1148	4.6614
0.9	9.7246	8.4851	11.0166	7.9510	High	12.8318
0.97	43.2670	28.5176	41.6309	27.7662	High	110.4259

۴- پیاده سازی سخت افزاری الگوریتم

همانطورکه گفته شد الگوریتم ارائه شده با دو واحد وزنی کار می کند. یکی سن سلول و دیگری تعداد سلولها در صف. این الگوریتم به صورت همزمان روی ستونهای ماتریس ورودی کار می کند که البته این مسئله ازطریق نرم افزار خیلی مشهود نیست. برای گرفتن کارایی بیشتر سخت افزاری به شکل زیر برای آن پیشنهاد می کنیم .



شکل ۲ : شمای سخت افزاری الگورتم WFMC

این سخت افزار شامل یک واحد محاسبه ماتریس سن، یک واحد محاسبه ماتریس طول صف ، یک واحد کنترل شمارش ، یک واحد محاسبه ماتریس وزن (جمع کننده) و واحد محاسبه ماتریس تطابق داورهای ۱ تا n می باشد. طرز کار مدار پیشنهادی بصورت زیر است.

واحد کنترل شمارش یکی از مهمترین واحد ها می باشد این واحد درصورتی که ماتریس ورودی برای درایه ای خاص ورودی داشته باشد در اینصورت سیگنال شمارش به بالا را به شمارنده های متناظر واحد محاسبه طول می فرستد. یکی از ورودیهای واحد کنترل شمارش ماتریس تطابق است با خروج یک درایه از ماتریس واحد کنترل شمارش سیگنال پایین شمار را به شمارنده متناظر در واحد محاسبه طول می فرستد. این واحد سه سیگنال کنترلی به شمارنده های واحد محاسبه سن می فرستد که عبارتند از سیگنال رست و سیگنال شمارش به بالا و

پایین . هنگامی که یک درایه ماتریس طول صفر شود در اینصورت سیگنال ریست به شمارنده متناظر فرستاده می شود و سن را صفر می کند. هنگامی که درایه ای از ماتریس تطابق ۱ شود و درایه نظیر آن در ماتریس طول صفر نباشد سیگنال پایین شمار را به شمارنده متناظر می فرستد. و هنگامی که درایه ای از ماتریس طول مخالف صفر باشد و درایه نظیر در ماتریس تطابق یک نباشد ، سیگنال بالا شمار به شمارنده متناظر در ماتریس سن می فرستد.

واحد جمع کننده درایه های نظیرماتریس سن و طول را با هم جمع می کند تا ماتریس وزن را محاسبه کند. ماتریس وزن به واحد محاسبه ماتریس تطابق فرستاده می شود و داوران روی ستونهای ماتریس جستجوکرده بزرگترین عنصر را پیدا نموده و فلگ مربوطه به سطر پیدا شده را یک می کنند. ماتریس تطابق پیدا شده به ساختار سویچ داده می شود و ساختار بر اساس آن پیکر بندی می شود. در ابتدای هر قطعه زمانی فلگ ها و ماتریس تطابق صفر می شوند.

۵- منابع

[۱] م.فتحی و ر. سعیدی نیا ، «ارائه یک الگوریتم زمانبندی برای سوییچهای سلولی با صف ورودی » در **مجموعه مقالات اولین کنفرانس بین المللی فن آوری اطلاعات و دانش ،**

[2] I.Keslassy, M.Kodialam, T.V.Lakshman, D.Stiliadis,"On Guaranteed Smooth Scheduling For Input-Queued Switches", IEEE Infocom 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE, Volume: 2, 30 March - 3 April 2003

[3] V.Tabatabaee, L.Tassiulas,"MNCM a new class of efficient scheduling algorithms for input-buffered switches with no speedup", IEEE Infocom 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE, Volume: 2 , 30 March - 3 April 2003

[4] S.Sarkar," Optimum Scheduling and memory management in Input Queued switches with finite buffer space", IEEE Infocom 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE, Volume: 2, 30 March - 3 April 2003

[5] M.Andrews, M.Vojnovic, "Scheduling Reserved traffic in Input-Queued Switches: New delay bounds via probabilistic techniques", IEEE Infocom 2003 Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE, Volume: 2, 30 March - 3 April 2003

[6] Paolo Giaccone, "Queuing and scheduling algorithms for high performance routers" , PHD thesis , 2002 , University of Politecnico DI Torino

[7] N.Mckeown, "Scheduling algorithms for Input-Queued Cell switches" , Ph.D. Thesis, 1995, University of California at Berkeley

[8] M. Ajmone Marsan, A. Bianco, P. Giaccone, E. Leonardi, F. Neri," Packet Scheduling in Input-Queued Cell-Based Switches", IEEE Infocom 2001,pp.1085-1094